

**Entwurf und Implementierung innovativer Architekturen für
Learning Content Management Systeme (LCMS) zum Einsatz
im erlösorientierten Bildungsexport**

Dissertation

Zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt der Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau

von Dipl.-Inf. Alexander Fleischer
geboren am 25.11.1975 in Suhl

Tag der Einreichung: 01.04.2008

Tag der wissenschaftlichen Aussprache: 07.11.2008

Gutachter: 1. Prof. Dr.-Ing. habil. W. Fengler, Technische Universität Ilmenau
 2. Dr.-Ing. H.-D. Wuttke, Technische Universität Ilmenau
 3. Prof. Dr.sc.techn. R. Ubar, Tallinn Technical University

urn:nbn:de:gbv:ilm1-2008000268

Zusammenfassung

Das Ziel dieser Arbeit ist die systematische Analyse der Anforderungen und die Erläuterung verschiedener Ansätze zur Realisierung dieser Anforderungen bei dem Einsatz von Learning Content Management Systemen (LCMS) im erlösorientierten Bildungsexport. Nach der systematischen Aufstellung der Anforderungen werden verschiedene LCMS hinsichtlich dieser Anforderungen analysiert und für die weitere Betrachtung eines der Systeme ausgewählt. Anhand dieses Systems wird die Realisierung weiterer Anforderungen, wie automatischer Datenaustausch, Einbindung von Multimedia und verschiedene Sicherheitsanforderungen dargestellt, welche im Weiteren auch als Prototyp realisiert wurden und im letzten Teil der Arbeit dargestellt sind.

Abstract

The objective of this work is the systematic analysis of the requirements and the clarification of different approaches of these requirements for the use of Learning Content Managemant Systems (LCMS) in revenue-oriented educational export. After the systematic specification of the requirements are different LCMS analyzed in terms of these requirements, and for the further consideration one of the systems selected. Using this system, the realization of other requirements, such as automatic data exchange, integration of multimedia and various security requirements presented, which in addition also as a prototype has been realized in the last part of the work presented.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fachgebiet Rechnerarchitektur des Institutes Technische Informatik und Ingenieurinformatik der Technischen Universität Ilmenau. Mit diesem vorangestellten Kapitel möchte ich allen danken, die zum Zustandekommen der Arbeit maßgeblich beigetragen haben.

An erster Stelle möchte ich dem Leiter des Fachgebietes Prof. Dr.-Ing. habil. Wolfgang Fengler danken. Er hat im mehrfachen Sinne dazu beigetragen, dass diese Arbeit zustande kam. Nach dem Abschluss meiner Diplomarbeit an seinem Fachgebiet ermöglichte er mir, im Projekt “Deutschsprachige Ingenieurinformatikausbildung am Moskauer Energetischen Institut” mitzuarbeiten. Im Rahmen dieses Projektes entstanden viele Ideen für diese Arbeit. Schließlich ermöglichte er mir die Anfertigung dieser Dissertation parallel zu meiner Tätigkeit im Projekt. Er ließ mir stets die benötigte Unterstützung zukommen und gewährte mir bei der Bearbeitung einen großzügigen Freiraum.

Besonders möchte ich mich bei Dr.-Ing. Heinz-Dietrich Wuttke - meinem zweiten Gutachter - bedanken, welcher mir mit zahlreichen Hinweisen zur Seite stand. Gleichzeitig möchte ich mich bei meinem dritten Gutachter, Prof. Ubar von der Universität Tallin in Estland bedanken.

Desweiteren möchte ich meinen Kollegen Dr. Bernd Däne, Dr. Jürgen Nützel, Dr. Olga Fenlger, Dipl.-Ing. Klaus-Dieter Fritz und Dipl.-Ing. Torsten Paschke sowie den Mitarbeitern des “Kompetenzzentrums E-Learning Dienste” der TU Ilmenau danken, die mir immer mit Rat und Tat zur Seite standen.

In dieser Arbeit sind die praktischen Ergebnisse einiger der von mir betreuten Hauptseminare, Studien- und Diplomarbeiten eingeflossen. Den Auftakt bildet die Diplomarbeit von Pavel Rakowskiy, welcher in seiner Arbeit eine erste Schnittstelle zum Durchsuchen der Daten eines LCMS implementierte. Alina Ageeva untersuchte in Ihrer Diplomarbeit die Übertragung von Lehreinheiten über Content-Syndication Protokolle. Ein System zum Schutz von Inhalten auf Java-Basis mittels asymmetrischer Verschlüsselung entwickelte Alexey Tselishchev in seiner Diplomarbeit. Herr Zhang Hao untersuchte im Rahmen eines Hauptseminars die

Steuerungs- und Scriptingeigenschaften des Windows Media Players. Mikhail Tarasov untersuchte schließlich in seiner Diplomarbeit den Austausch von umfangreichen Daten mit Hilfe von Peer-to-Peer Protokollen. Bei diesen Studenten möchte ich mich ebenfalls bedanken.

Schließlich gilt mein Dank den Entwicklern der verwendeten Open Source Software, ohne Sie wäre diese Arbeit nicht möglich gewesen. Aufgrund des hohen Anteils an Open Source Software ist naturgemäß der Anteil der referenzierten Internet-Quellen relativ hoch.

Inhaltsverzeichnis

1	Motivation und Einleitung	1
2	Stand der Technik	4
2.1	Content Management Systeme	4
2.2	Learning Management Systeme	7
2.3	Learning Content Management Systeme (LCMS)	7
2.4	Protokolle zum Datenaustausch	22
2.5	Standards zum Austausch von Lehrinhalten	27
2.6	Technologien für die Präsentation interaktiver Lehrinhalte	32
2.7	Online Bezahlssysteme	34
2.8	Sicherheit und rechtliche Aspekte	36
3	Anforderungen an e-Learning Systeme für den Einsatz im erlösorientierten Bildungsexport	41
3.1	Anforderungen an das LCMS	42
3.2	Sicherheitsanforderungen	48
3.3	Technische Anforderungen an Lehrinhalte	52
3.4	Datenaustausch mit anderen Systemen	52
4	Möglichkeiten zur Realisierung der Anforderungen	54
4.1	LCMS	54
4.2	Sicherheit und rechtliche Aspekte	62
4.3	Interaktive Inhalte und Anwendungen	64
4.4	Formate und Formatkonvertierungen	68
4.5	Multimediaanbindung	70
4.6	Datenaustausch	72
5	Anwendungsbeispiele	83
5.1	Schnittstelle zum Austausch von Metadaten	83

5.2	Synchronisation von Inhalten zwischen mehreren ILIAS-Systemen	85
5.3	Applet für interaktive Folien	87
5.4	Editor zum Erstellen der Applet - Beschreibungsdateien	97
5.5	Applets zur Simulation spezialisierter Inhalte	98
5.6	Paymentfunktionen in ILIAS	100
5.7	Playerscripting zur Synchronisation von Video- / Audiostreams und Präsentationsdaten	104
6	Zusammenfassung und Ausblick	107
	Literaturverzeichnis	109

Abbildungsverzeichnis

2.1	LCMS metacoon	11
2.2	LCMS ILIAS	15
2.3	LCMS moodle	17
2.4	Ablauf des Bezahlvorgangs bei verbreiteten HTTP-basierten Bezahlssystemen	35
4.1	Übersicht über die realisierten Komponenten	55
4.2	DRM in Java - Lizenzierung (aus [Tse06])	63
4.3	Datenaustausch mit zusätzlichem Server im entfernten Netz	74
4.4	Aufbau der Synchronisation über SOAP	80
5.1	Ablauf der Datensynchronisation	88
5.2	Applet für interaktive Folien	89
5.3	Applet für interaktive Folien: Klassendiagramm	91
5.4	Applet für interaktive Folien: Klassendiagramm Elementklassen	93
5.5	Editor für Applet-Beschreibungsdaten	98
5.6	Applet Petri Netz Simulator mit Erreichbarkeitsgraph	99
5.7	Ablauf des Bezahlvorgangs aus Benutzersicht	101

Tabellenverzeichnis

2.1	Vergleich der LCMS Systeme	23
4.1	Vergleich der LCMS Systeme bzgl. der Anforderungen	56

Quelltextverzeichnis

5.1	Abfrage der Metadaten über die SOAP-Schnittstelle in ILIAS	84
5.2	Auslagerung von Berechnungen auf den Server - Java-Applet (Client)	100
5.3	Auslagerung von Berechnungen auf den Server - serverseitige Berechnung . . .	101
5.4	Übergabe der Zahlungsdaten im Formular	102
5.5	Prüfung der Rückgabewerte nach Abschluss der Zahlung	103
5.6	Aufzeichnung der Folienwechselzeiten in Powerpoint	105

Abkürzungsverzeichnis

AICC	Aviation Industry CBT Committee
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CMS	Content Management Systeme
CORBA	Common Object Request Broker Architecture
DRM	Digital Rights Management
HTTP	Hypertext Transfer Protocol
ICE	Information and Content Exchange
IMS QTI	IMS Question & Test Interoperability
LCMS	Learning Content Management Systeme
LMS	Learning Management Systeme
LOM	Learning Objects Metadata
RPC	Remote Procedure Call
RSS	Really Simple Syndication
SCORM	Sharable Content Object Reference Model
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TLS	Transport Layer Security
UDDI	Universal Description, Discovery and Integration
WCMS	Web Content Management Systeme
WSDL	Web Service Description Language
XML	Extensible Markup Language

Kapitel 1

Motivation und Einleitung

Im Rahmen des Projektes “Deutschsprachige Ingenieur fakultät im Moskauer Energetischen Institut” wurden Mittel und Wege benötigt, Lehrinhalte auch ohne den Austausch von Lehrkräften den Studenten der Partneruniversität zu vermitteln. Die Lösung dafür findet sich im E-Learning.

Dieses Projekt stellte damit den initialen Ansatz zu dieser Arbeit dar. Im folgenden wurden systematisch Anforderungen an e-Learning Systeme aufgestellt (siehe Kapitel 3), welche im Rahmen des Bildungsexportes benötigt werden. Insbesondere wurde hier Wert gelegt auf einfachen Datenaustausch zwischen mehreren Einrichtungen und verschiedene Sicherheitsanforderungen. Auch wurden bei der Aufstellung der Anforderungen verschiedene technische Rahmenbedingungen, wie beispielsweise die Internetanbindung des Partners, berücksichtigt. Es wurden dabei nur Open Source Systeme untersucht, da es in diesem Bereich bereits mehrere Open Source Systeme mit unterschiedlichen Charakteristika existieren, keine Lizenzkosten für die Nutzung des Systems anfallen und diese Systeme durch das Vorliegen des Quelltextes auch einfach an weitere Anforderungen anpassbar sind. Das zu realisierende System soll frei verfügbar und erweiterbar sein, und eine Synchronisation der Inhalte mit anderen Systemen erlauben. Anhand der Anforderungen wurden verschiedenen Learning Content Management Systeme verglichen und eines der Systeme ausgewählt.

Für Anforderungen, die das Learning Content Management System (LCMS) allein nicht realisieren konnte, wurden verschiedene Realisierungsvarianten betrachtet. Dabei wurden bei einigen Anforderungen mehrere Möglichkeiten dargestellt, welche im späteren Einsatz parallel verwendet werden können, beispielsweise bei der Einbindung von Multimediainhalten, bei anderen Anforderungen erfolgte eine Auswahl der zu verwendenden Technologie für die spätere Realisierung. Diese erfolgte im Rahmen eines Prototyps für das ausgewählte LCMS.

Die Einbindung von Multimediainhalten in interaktive Lernmaterialien ist eine dieser Anforderungen, da die Studenten parallel zur fachlichen Ausbildung auch ihre Kenntnisse in Deutsch als Fachsprache vertiefen sollen, und dies die Einbindung von gesprochenen Texten und Videos erfordert. Diese Anforderungen entstanden bei der Erstellung von Lehrmaterialien im Rahmen des Projektes, in welchem ca. 50 interaktive Folien mit Audiounterstützung, mehrere unabhängige Applets für spezielle Simulationen, einige Vorlesungen und Seminare als Videoaufzeichnung und ca. 200 Seiten mit begleitenden Texten entstanden.

Es sollte ebenfalls untersucht werden, wie sich LCMS Systeme zur Verwaltung von bezahlten Inhalten eignen und wie man diese vor unbefugtem Zugriff schützen kann.

In Kapitel 2 wird der Stand der Technik dargestellt. Es werden Content Management Systeme und Learning Management Systeme betrachtet, auf welche die im folgenden betrachteten Learning Content Management Systeme (LCMS) aufbauen. Dabei werden dort verschiedene Open Source LCMS detailliert betrachtet und der jeweilige Funktionsumfang dargestellt. Weiterhin werden in Kapitel 2 einige Basistechnologien für die später realisierten Erweiterungen dargestellt. Dabei werden Datenaustauschprotokolle, Techniken zur Präsentation interaktiver Lehrinhalte, Möglichkeiten zur direkten Bezahlung von Inhalten und weitere Sicherheitsaspekte betrachtet.

Im Kapitel 3 wird systematisch eine Liste mit Anforderungen aufgestellt, welche ein LCMS für den Einsatz im erlösorientierten Bildungsexport erfüllen sollte. Dabei wird sowohl auf Anforderungen an das LCMS selbst, wie auch auf die Inhalte, Sicherheitsanforderungen und Möglichkeiten zum Datenaustausch mit anderen Systemen eingegangen.

Im Kapitel 4 werden verschiedene Möglichkeiten betrachtet, diese Anforderungen zu realisieren. Anhand der Anforderungen werden die bereits im Kapitel 2 dargestellten LCMS verglichen, und anhand dieses Vergleichs ein System ausgewählt, welches die meisten der gestellten Anforderungen erfüllt. Ebenfalls werden dort verschiedene Möglichkeiten zur Realisierung interaktiver Inhalte dargestellt. Diese sind größtenteils unabhängig vom eingesetzten LCMS nutzbar. Es wird untersucht, welche Technik sich für welche Aufgabenstellung am besten eignet. Anhand dieser Darstellung ist es möglich, die jeweils für die zu erstellenden Anwendungen passende Technik zu wählen, da diese ohne Probleme auch parallel verwendet

werden können. Im Weiteren erfolgt die Betrachtung der nötigen Erweiterungen des LCMS, um die aufgestellten Anforderungen zu erfüllen. Neben der Konvertierung von verschiedenen Formaten werden hier verschiedenen Möglichkeiten zur Synchronisation von mehreren LCMS Systemen über verschiedene Medien und Bandbreiten dargestellt und analysiert. Weiterhin werden in Kapitel 4 noch die Einbindung von Multimediaelementen wie Audio und Video in interaktive Lerneinheiten und Sicherheitsaspekte dargestellt.

Im Kapitel 5 wird die Umsetzung von ausgewählten Komponenten der in Kapitel 4 dargestellten Varianten als Erweiterung des LCMS ILIAS dargestellt. Ebenfalls werden hier verschiedene interaktive Inhalte und die Einbindung von Multimediaelementen in Lerneinheiten dargestellt.

Kapitel 2

Stand der Technik

In diesem Kapitel werden die Systeme und Technologien, welche in der Arbeit betrachtet wurden, vorgestellt. Neben einer Auswahl von Open Source Learning Content Management Systemen werden Protokolle zum Datenaustausch, Techniken zur Präsentation interaktiver Inhalte, Bezahlssysteme und verschiedene Sicherheitsaspekte betrachtet.

2.1 Content Management Systeme

Content Management Systeme (CMS) sind Systeme zur Verwaltung von Inhalten. Es gilt dabei das Prinzip der Trennung von Inhalt, Design und Funktion. CMS speichern die Inhalte in einem medienneutralen Datenformat wie zum Beispiel XML und erzeugen das gewünschte Ausgabeformat, wie z.B. HTML oder PDF, zur Laufzeit dynamisch. CMS soll auch Benutzern ohne Kenntnisse über die verwendeten Technologien ermöglichen, Inhalte zu erstellen, zu bearbeiten und zu verwalten. Das CMS übernimmt neben der Umwandlung in verschiedene Anzeigeformate auch die Erstellung einer geeigneten Navigationsstruktur, stellt Suchfunktionen bereit, ermöglicht die Erfassung von Metadaten zu den einzelnen Inhalten, und verwaltet Zugriffsrechte für die einzelnen Inhalte. Viele CMS bieten auch Funktionen zum Dokumenten-Management an, d.h. im CMS können auch Dokumente in fremden Dateiformaten, wie z.B. Word-Dokumente, aufgenommen und verwaltet werden.

Ein Teilbereich der Content Management Systeme bilden die Web Content Management Systeme (WCMS, [Fuc07]). Diese sind darauf ausgerichtet, die Inhalte im World Wide Web zu präsentieren. Die Speicherung und Verwaltung der Inhalte erfolgt in der Regel mit Hilfe einer Datenbank auf dem Server, auch die meisten Teile der Programmlogik werden direkt auf dem Server ausgeführt. Daher gehören die meisten WCMS zur Gruppe der serverbasierten

CMS. Als Client zum Anlegen, Bearbeiten und Verwalten von Inhalten, Konfiguration der Darstellung und allen weiteren Funktionen des CMS wird daher nur ein normaler Webbrowser benötigt, eine Installation einer speziellen Software ist nicht erforderlich.

Es gibt zahlreiche WCMS auf dem Markt, neben kommerziellen Systemen sind auch zahlreiche freie Programme verfügbar. Eine Auswahl dieser Systeme findet man auf der Übersicht unter [Age]. Ein Beispiel für ein freies WCMS ist das an der TU Ilmenau zur Verwaltung der Webseiten eingesetzte System Typo3 ([AFH06], [TYP]). Dieses serverbasierte WCMS hat folgende Funktionen, welche in ähnlicher Ausprägung auch in anderen WCMS vorhanden sind:

- **Verwaltung von Inhalten** Typo3 legt die Inhalte in einer zentralen Datenbank auf dem Server oder im Dateisystem zentral auf dem Server ab. Die Inhalte können mit Metadaten wie Autor und Kurzbeschreibung versehen und in mehreren Sprachen erfasst werden.
- **Erstellung der Navigationsstruktur** Typo3 erstellt automatisch eine entsprechende Navigationsstruktur, bestehend aus hierarchischen, grafischen Menüs, lokalen Menüs, Anzeigen des aktuellen Pfades und weiteren, konfigurierbaren Elementen.
- **Indizierung und Suche** Die gespeicherten Inhalte werden indiziert und können über eine Volltextsuche erschlossen werden.
- **Rendering für verschiedene Ausgabemedien** Typo3 kann die Inhalte für verschiedene Ausgabemedien, wie Beispiel HTML für Webbrowser, PDF zum Ausdruck oder WML zur Darstellung auf mobilen Geräten rendern. Dabei können jeweils verschiedene Vorlagen und Einstellungen benutzt werden.
- **Autorenoberfläche** Mit Hilfe des *Backends*, einer einfach zu bedienenden Autorenoberfläche, können neue Inhalte erstellt und vorhandene Inhalte bearbeitet werden. Dabei werden verschiedene Inhaltstypen wie formatierter Text (welcher direkt im Browser mit einer Word-ähnlichen Oberfläche erstellt werden kann), Tabellen, Bilder und Multimediaelemente wie Flash oder Videos unterstützt.
- **Rechteverwaltung** Die Rechtevergabe im *Backend* von Typo3 ist dem Rechtesystem

in UNIX-artigen Betriebssystemen ähnlich. Für jede Seite und jeden Inhalt kann für Besitzer, Gruppe und Andere festgelegt werden, wer die entsprechenden Inhalte bearbeiten, löschen und freigeben darf. Der Administrator kann außerdem für jede Benutzergruppe einen Teilbaum der Inhalte bestimmen, welche diesem angezeigt werden, und einzelne Funktionen für bestimmte Benutzergruppen ausblenden. Zusätzlich können getrennt dazu die Rechte für das *Frontend*, also die erstellten Ausgabeseiten, eingestellt werden. Neben der Konfiguration für verschiedene, vom Backend unabhängige Nutzergruppen können Beschränkungen auch abhängig von IP Adresse oder anderen Bedingungen eingesetzt werden.

- **Administration** Die Administration findet ebenfalls im *Backend* statt. Über die eigene Scriptsprache *TypoScript* wird die entsprechende Konfiguration der Ausgabemedien, der Navigationsstruktur und der Darstellung vorgenommen. Für verschiedene Backend-Benutzergruppen ist es außerdem zusätzlich zur Rechteverwaltung möglich, die Ansicht und die Einstellungen des Backends mit Hilfe von Skripten (*TSconfig*) zu ändern. Ebenfalls im *Backend* können auch zusätzliche Funktionen des WCMS, welche in Form von Erweiterungen installiert werden können, verwaltet werden.
- **einfache Erweiterbarkeit** Typo3 kann Aufgrund des modularen Aufbaus sehr einfach um neue Funktionen, wie neue Inhaltstypen, Arbeitsabläufe, Editoren, etc. erweitert werden. Zur Erstellung von Erweiterungen gibt es eine Unterstützung in Form des *Kickstarter*, welcher nach Eingabe der nötigen Parameter einen Rahmenquelltext erzeugt, der bereits alle Funktionen für die Anpassung der Datenbank und die Einbindung im System enthält. Die neue Funktionalität kann dann entweder mit Hilfe von *TypoScript* oder direkt in der Programmiersprache *PHP* erstellt werden.
- **ausführliche Dokumentation** Für Typo3 existieren zahlreiche Handbücher in elektronischer Form, neben allen Funktionen für Autoren und Redakteure sind auch die Administration und die API für Erweiterungen sehr gut dokumentiert. Zusätzlich gibt es auch zahlreiche Demonstrationsvideos, welche das Vorgehen für einzelne Aufgaben in Bild und Ton erläutern.

2.2 Learning Management Systeme

Learning Management Systeme (LMS) sind Systeme zur Planung und Verwaltung von Lehrangeboten. Dazu gehört das Planen von Kursen, Erstellen von Lehrplänen, die Verwaltung der Kursteilnehmer (Anmeldung, Bezahlung von Kursen, Bereitstellung von Inhalten, Speicherung von Ergebnissen, Feststellung des Lernfortschritts, Speicherung von Kursabschlüssen, etc.), das Verwalten von Ressourcen wie z.B. Räumen und technischen Geräten, und weitere Funktionen.

Ein Beispiel für ein Learning Management System ist stud.ip ([dat05]), welches eine derartige Verwaltung von Lehrveranstaltungen auf der Basis eines Open Source Systems ermöglicht.

2.3 Learning Content Management Systeme (LCMS)

Learning Content Management Systeme (LCMS) sind eine Kombination aus CMS und LMS. Neben den Funktionen eines klassischen CMS, dem Erstellen, Wiederverwenden, Auffinden, Bearbeiten und Ausliefern von Inhalten werden auch die Funktionen eines LMS, wie das Zusammenstellen von Kursen und die Verwaltung von Kursteilnehmern unterstützt.

2.3.1 Verwaltung von Inhalten

LCMS benutzen in der Regel ein zentrales Repository, in welchem die Inhalte in Form von Lernobjekten, Abschnitten, Multimediadaten und weiteren Objekten abgelegt sind. Zu jedem Objekt können Metadaten, beispielsweise nach dem LOM Standard erfasst werden, um die einzelnen Objekte zu beschreiben und im Repository für eine Suche zugänglich zu machen. Die Lernobjekte können aus verschiedenen Kursen referenziert werden, so dass bei einer Änderung an einem Lernobjekt dieses automatisch in allen Kursen auf den aktuellen Stand gebracht wird. Der Zugriff auf die einzelnen Lernobjekte, Kurse und weitere Daten wird über eine interne Benutzer- und Rechteverwaltung reglementiert. Bei einigen LCMS ist auch eine Versionsverwaltung vorhanden, so dass Änderungen an Lernobjekten nachvollzogen werden können.

Um die Wiederverwendung der Inhalte zu ermöglichen, ist es möglich, Lernobjekte zu importieren und zu exportieren. Neben eigenen Im-/Exportformaten werden meist auch ein oder mehrere der in Kapitel 2.5 aufgeführten Standards, sowie einfache Datentypen wie Text oder HTML unterstützt.

2.3.2 Autorenfunktionen

Die meisten LCMS unterstützen genau wie CMS die direkte Erstellung von Inhalten in einem Online-Editor. Dabei ist es möglich, lerninhaltspezifische Informationen, wie zum Beispiel Metadaten oder semantische Informationen, mit anzulegen. Auch wird meist das Einbinden von komplexen Inhalten, wie Formeln, Java Applets, Audio und Video, aber auch Fragenkatalogen, Tests, Glossaren und Literaturverzeichnissen direkt unterstützt. Die Inhalte werden als einzelne Lernobjekte angelegt, welche durch Kapitel, Unterkapitel und Abschnitte weiter gegliedert werden. Diese können dann mit den LCMS Funktionen zu Kursen zusammengestellt werden.

Einige Lernsysteme erlauben es auch, die Inhalte mit einem komfortablen externen Programmsystem, wie beispielsweise dem Open Source Büropaket OpenOffice zu erstellen und diese Inhalte dann in das Format für Lerneinheiten zu konvertieren. Dabei müssen die einzelnen Inhalte semantisch gekennzeichnet werden, damit das LCMS die einzelnen Elemente, wie Absätze, Quelltext, Glossareinträge, Lehrsätze und Beispiele, entsprechend zuordnen und später in geeigneter Weise im Repository einbinden kann. Beispiele hierzu werden im Abschnitt 2.3.4 genannt.

2.3.3 Interaktive Wissenspräsentation

LCMS enthalten Komponenten zum Rendern der Inhalte für verschiedene Ausgabegeräte, mindestens jedoch für die Anzeige im Webbrowser. Folgende Funktionen werden von vielen LCMS unterstützt, um den Lernenden zu assistieren:

- Markieren einzelner Seiten als „bearbeitet“.
- Anlegen von Kommentaren und Lesezeichen zu einzelnen Seiten.

- Speicherung von Testergebnissen, Vergleich mit Musterlösungen.
- Kontextbezogene Kommunikationsmittel, wie z.B. Foren.

2.3.4 Vergleich aktueller LCMS

In den folgenden Abschnitten werden die wichtigsten Eigenschaften einiger Lernplattformen dargestellt und verglichen. Es wurden dabei nur frei verfügbare, unter einer Open Source Lizenz stehende Plattformen betrachtet, da es nur dort möglich ist, die Plattformen bei Bedarf an eigene Anforderungen anzupassen und zu erweitern.

2.3.4.1 Metacoön

Metacoön (Abbildung 2.1) wurde an der Bauhaus Universität Weimar entwickelt und stellt eine Lern- und Kommunikationsplattform bereit. Die Plattform unterteilt sich dabei in verschiedene Räume. Neben einem allgemeinen Eingangsraum, auf welchen alle Plattformmitglieder Zugriff haben, können beliebig viele Arbeitsräume für verschiedene Bereiche oder Veranstaltungen angelegt werden. In den einzelnen Räumen stehen zahlreiche Module zum Lernen und zur Kommunikation zur Verfügung, die über ein komplexes Rechtesystem in den jeweiligen Räumen verwaltet werden können. Es existieren folgende Module:

- Kommunikation
 - Chat
PHP / JavaScript basiertes Chatsystem
 - Diskussionsforen
3 stufiges Forensystem, bei allen Einträgen können auch beliebige Dateien angehängt werden
 - Pinnwand
Pinnwandsystem, welches durch die Möglichkeit, dort Dateien, Grafiken, Verweise, Foren und Umfragen anzuhängen sehr flexibel ist
 - Adressbuch, Mailverteiler und Visitenkarten zur Verwendung mit externen Mailprogrammen

- Sofortnachrichten

Sofortnachrichten dienen der direkten Kommunikation untereinander und werden, sofern der Empfänger der Nachricht gerade online ist, sofort zugestellt, ansonsten bis zum nächsten Anmelden des Empfängers zwischengespeichert. Während in früheren Versionen von metacoon für diese Funktionalität ein Java-Applet zum Einsatz kam, wird dies in der aktuellen metacoon-Version per AJAX realisiert.

- Dateimanager

allgemeines hierarchisches Dateiverzeichnis zur Verwaltung und zum Austausch beliebiger Dateien

- Lernmodule

- Lehrmaterial-Pool

durchsuchbares Verzeichnis der verfügbaren Lehrmaterialien, analog zur Pinnwand können auch hier Dateien, Verweise, Foren und Umfragen angehängt werden

- Kursverzeichnis

Verzeichnis verfügbarer Kurse und Zusammenstellung neuer Kurse aus extern erstellten Lehrmaterialien

- Veranstaltungsverwaltung

einfaches Einschreibungssystem

- Nachschlagewerke (Glossar, Projekte, Personen und weitere)

- Literaturverzeichnis

- Weblinkverzeichnis

Weiterhin können extern mit Hilfe eines OpenOffice Plugins erstellte Lehrmaterialien sowie SCORM-Lehrmaterialien eingebunden werden. Mit Hilfe eines QTI-Editors auf Java-Basis und des dazugehörigen Players auf der Basis von Flash ist es möglich, Tests zu realisieren. Darüber hinaus erlaubt metacoon die Pflege eines zentralen Repositories mit Hilfe eines Java-Programms, welches auch das Anlegen und Durchsuchen beliebiger Metadaten auf XML-Basis zu den einzelnen Einträgen erlaubt.

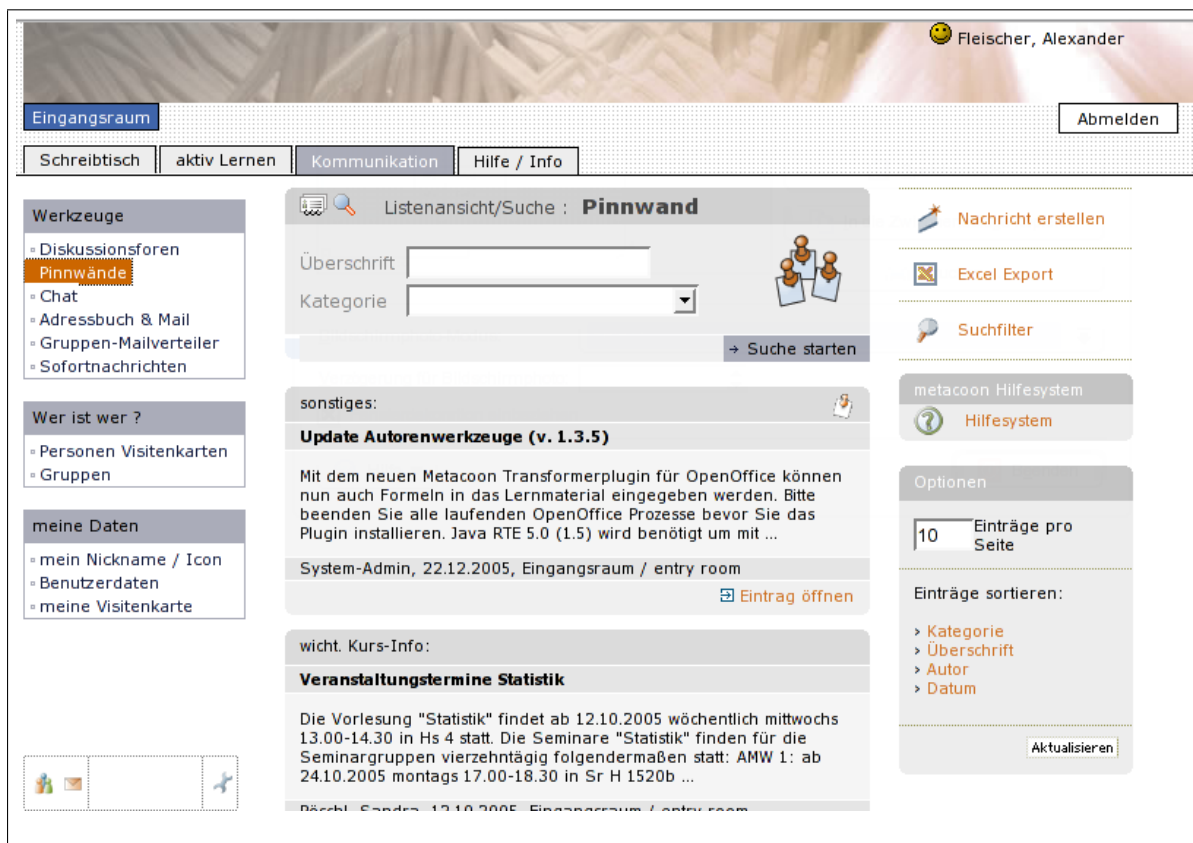


Abbildung 2.1: LCMS metacoon

Technische Realisierung Die Grundfunktionen der metacoon-Plattform werden mit Hilfe von PHP-Skripten und MySQL als relationale Datenbank realisiert. Die Daten werden größtenteils in der Datenbank, teilweise aber auch im Dateisystem des Servers verwaltet. Zahlreiche Funktionen werden clientseitig per JavaScript realisiert, so dass ein moderner Browser zum Benutzen der Plattform notwendig ist.

Die Rechteverwaltung für einzelne Objekte ist an das Modell von Unix angelehnt, zu jedem Objekt können Rechte für eine bestimmte Gruppe und für „Alle“ vergeben werden. Es existieren weder lokale Benutzergruppen noch gibt es die Möglichkeit, dass Gruppen wieder Mitglieder von Gruppen sind, was die Verwaltung der Rechte etwas einschränkt. Zusätzlich existiert eine Rechteverwaltung für jeden einzelnen Raum, über welche die Verfügbarkeit und Funktionalität der verschiedenen Module an die jeweilige Benutzergruppe angepasst werden kann.

Die Benutzer können auf verschiedene Arten authentifiziert werden, neben der internen Au-

thentifizierung steht z.B. auch ein Modul zur Anbindung an einen LDAP-Server zur Verfügung. Eine Erweiterung der Authentifizierungsmechanismen ist dabei relativ einfach möglich.

Für das Wiedergeben von extern erstellten Kursen und SCORM-Lerneinheiten wird Cocoon ([Nie06]) zur Transformation der Daten eingesetzt. Die Kurse können dabei aus den einzelnen Kapiteln / Bestandteilen in einem Kursverzeichnis zusammengestellt werden, diese werden dann von Cocoon gerendert und über einen Link im Lehrmaterial-Pool zur Verfügung gestellt.

metacoocn enthält keine Online-Autorenwerkzeuge, alle Inhalte müssen extern erstellt werden. Für die Erstellung von Lehrmaterialien existiert dazu ein Plugin für OpenOffice, welches ein mit speziellen Absatzformaten versehenes OpenOffice Dokument in eine XML-Datei transformiert und zusammen mit dem Originaldokument, Metadaten und nötigen Grafiken / Applets / Multimediadateien in ein Archiv verpackt. Diese Archive können dann online zu Kursen zusammengestellt werden. Zusätzlich ist hier auch die Einbindung von Fragen möglich, welche mit einem externen Programm im QTI Format (siehe Kapitel 2.5.5) erstellt werden können.

2.3.4.2 ILIAS

Das von der Universität Köln seit dem Jahr 2000 entwickelte LCMS ILIAS open source (Abbildung 2.2, [ILI]) benutzt ein hierarchisches System zur Verwaltung von Lehrinhalten. Mit Hilfe von Kategorien und Ordnern ist es möglich, die einzelnen Lerninhalte sehr übersichtlich zu strukturieren. ILIAS kennt unter anderem folgende Elemente:

- Lernmodule

- ILIAS Lernmodule

ILIAS Lernmodule können direkt online erstellt werden und dienen der Präsentation von Lerninhalten. Eine Lerneinheit besteht aus beliebig vielen Abschnitten, Unterabschnitten und Seiten. Jede Seite kann aus verschiedenen Inhaltselementen, wie Überschriften, Absätzen, Tabellen, Listen, Grafiken, Multimediaelementen und Dateiverweisen zusammengestellt werden. Es ist außerdem möglich, die einzelnen Inhaltselemente semantisch zu kennzeichnen, beispielsweise als Definition, Merksatz, Beispiel oder zusätzliche Information. Zu jedem Element können Metadaten

nach dem LOM Standard erfasst werden. Die Erstellung der Lerneinheiten in verschiedenen Sprachen ist möglich, die Sprache der jeweiligen Lerneinheit wird in den Metadaten angegeben.

- Lernmodule in anderen Formaten

ILIAS erlaubt das Einbinden von Lernmodulen im HTML, AICC oder SCORM Format. ILIAS unterstützt dabei den SCORM 1.2 Standard vollständig.

- Fragenpool und Test

Es können Tests angelegt und direkt online bearbeitet werden. Die Fragen dazu lassen sich zur einfacheren späteren Wiederverwendung in Fragenpools organisieren. Neben einfachen „Multiple Choice Fragen“ werden auch Zuordnungsfragen, Lückentexte und Java-Applets unterstützt. Die Tests können als Prüfungsvorbereitung und zur Selbstkontrolle, aber auch als Zugangsvoraussetzung für andere Lernmodule verwendet werden. Die erstellten Fragenpools/Tests können im standardisierten QTI - Format für die Verwendung in anderen Anwendungen exportiert werden.

- Übungen

Mit Hilfe des Moduls Übung können auf einfache Art und Weise Aufgaben gestellt werden. Dabei ist es möglich, die Aufgaben verschiedenen Gruppen zuzuordnen und Dateien anzuhängen. Die Studenten können ihrerseits dann die Lösungen als Datei zur Kontrolle hochladen.

- Ordner und Dateien Lehrmaterialien in anderen Formaten oder allgemein Dateien können sehr einfach in die bestehende Struktur eingebunden werden.

- Recherchemodule

Es stehen eine Linkverwaltung mit automatischer Prüfung und ein Glossar zur Verfügung.

- Kurse

Kurse sind ein Container für die Zusammenstellung der Lernmaterialien und Kommunikationsmodule. Dabei ist es hier möglich, einzelne Materialien auch zu verlinken, so dass das gleiche Material in mehreren Kursen verwendet werden kann. Trotzdem ist die Erstellung individuelle Übungen und Tests möglich. Die einzelnen Materialien können geordnet werden, und sind auch zeitgesteuert aktivierbar.

Kurse dienen gleichzeitig der Organisation der Teilnehmer. Teilnehmer können einem Kurs beitreten und als Kursmitglied erweiterten Zugriff erhalten. Dabei gibt es hier verschiedene Verfahren, neben dem direkten Beitritt ist es auch möglich, das ein Beitritt nur bei Kenntnis eines Kurspasswortes möglich ist oder auch nur nach Bestätigung durch einen Kurs-Administrator.

- Kommunikationsmodule

- Chat

Das Chatmodul in ILIAS basiert auf HTTP, setzt aber zur Ausgabe des Datenstroms einen eigenen Java-Server auf der Serverseite voraus.

- Foren

ILIAS ermöglicht das Anlegen hierarchisch aufgebauter Foren.

- Terminkalender

Der eingebaute Terminkalender kann für verschiedene Gruppen und Kurse gemeinsame Termine verwalten. Entsprechende Gruppentermine werden den Teilnehmern angezeigt und zur Übernahme in ihren persönlichen Kalender vorgeschlagen, auch spätere Änderungen von Gruppenterminen werden mit erfasst.

- Gruppen

Im ILIAS ist es möglich, Gruppen zu bilden und Lernmaterialien, Aufgaben und natürlich Mitglieder in diesen Gruppen zu organisieren. Eine Gruppe ist dabei ein Containerobjekt, welches beliebige andere Objekte enthalten kann.

- Sessions

Sessions wurden mit ILIAS 3.7 eingeführt und erlauben das Anlegen und Zuweisen von Materialien und Teilnehmern zu bestimmten Terminen, welche Session genannt werden.

Autorisierung - Rechteverwaltung Die Rechteverwaltung in ILIAS ist hierarchisch aufgebaut und rollenbasiert, neben allgemeinen globalen Rollen ist es möglich, weitere, lokale Rollen für einzelne Kategorien und Kurse zu definieren und für Teilbäume damit ergänzende Rechte zuzuweisen. Diese Rechteverwaltung erlaubt eine sehr detaillierte Einstellung der erlaubten Aktionen. Für einige Objekte werden automatisch Rollen erstellt und verwaltet, wie

The screenshot displays the ILIAS LCMS interface for a user named Alexander Fleischer. The top navigation bar includes links for 'Persönlicher Schreibtisch', 'Magazin', 'Suchen', 'Mail', and 'Administration'. The main content area is titled 'Persönlicher Schreibtisch' and features a sidebar with tabs for 'Übersicht', 'Persönliches Profil', 'Kalender', 'Private Notizen', 'Bookmarks', and 'Lernfortschritt'. The main content is organized into several sections: 'Ausgewählte Angebote' (Selected Offers) with a 'Details ausblenden' button, 'Kategorien' (Categories) with items like 'Rechnerarchitekturen 2 - Zusatzseminar II' and 'Rechnernetze der PDV', 'Kurse' (Courses) with 'Praktikum TI2 - 1' and 'Praktikum TI2 - 2', and 'Foren' (Forums) with 'CAN Praktikum' and 'Forum'. Each item has 'Bearbeiten' (Edit) and 'Vom Schreibtisch nehmen' (Remove from Desktop) buttons. On the right, there are sidebars for 'Aktive Benutzer' (Active Users) showing '1 registrierter Benutzer' (1 registered user) and 'Meine Bookmarks' (My Bookmarks) showing 'Es sind keine Bookmarks angelegt.' (No bookmarks created).

Abbildung 2.2: LCMS ILIAS

beispielsweise für Kurs-Mitglieder oder Foren-Administratoren. Diesen können wiederum beliebige Benutzer zugeordnet werden, so dass eine Konfiguration nicht nur global, sondern auf der Ebene des einzelnen Objektes möglich ist.

Mit Hilfe dieser detaillierten Konfigurationsmöglichkeiten ist es beispielsweise auch möglich, dass Teilnehmer in bestimmten Gruppen selbst das Recht haben, Inhalte zu erstellen und zu bearbeiten.

Autorenwerkzeuge ILIAS bietet die Möglichkeit, viele Arten von Lernmaterialien direkt auf der Plattform online zu erstellen. Es ist dabei möglich, häufig benötigte Mediadateien im "Media Pool" zu organisieren, vorhandene Materialien anzupassen und zu verändern, als auch natürlich neue Materialien zu erstellen. Zur Erstellung von textuellen Inhalten kommt dabei

ein Rich-Text Editor zum Einsatz, der die semantische Auszeichnung des Inhaltes unterstützt.

Neben der Möglichkeit, Inhalte direkt auf der Plattform zu erstellen, existiert auch noch die Möglichkeit, Inhalte vorab mit Hilfe des OpenOffice Paketes zu erstellen. Für OpenOffice existiert dazu das Plugin iLAIX [Bol], welches die nötigen Vorlagen zur Auszeichnung und den Export ins ILIAS-Format bereitstellt.

technische Realisierung ILIAS wurde überwiegend in der Programmiersprache PHP entwickelt. Zur Datenbankabstraktion wird PEAR::DB bzw PEAR::MDB2 verwendet, so das ILIAS nicht nur ausschließlich mit MySQL als relationaler Datenbank, sondern auch mit anderen durch diese Module unterstützten Datenbanken zusammenarbeitet.

ILIAS verwendet intern eine globale Baumstruktur, die alle Objekte des LCMS enthält. Diese wird mit Hilfe des Nested Set Modells in der Datenbank abgelegt und gepflegt. Die inneren Knoten des Baumes bilden dabei Containerobjekte wie Kategorien, Kurse oder Gruppen, welche dann jeweils neben anderen Containerobjekten auch alle anderen Objekttypen enthalten können. Dabei ist es möglich, Verknüpfungen zu erstellen und ein Objekt (beispielsweise ein Lernmodul) in mehreren Containern zu verknüpfen. Damit wirken sich Änderungen an diesem Objekt auf alle Instanzen des Objektes aus. Benutzerrechte und Rollen werden innerhalb der Baumstruktur vererbt, so das jeweils nur neue, geänderte Einstellungen ergänzt werden müssen, wenn ein neues Objekt angelegt wird.

ILIAS Lernmodule folgen ebenfalls diesem Aufbau, auch eingebundene Multimediaobjekte können an verschiedenen Stellen referenziert werden. Im Fall von Java-Applets ist es dabei auch möglich, jeweils verschiedene Parameter für jede Instanz vorzugeben, falls das Applet beispielsweise die Anzeige / Simulation von verschiedenen Objekten beherrscht.

Die Benutzer können sich über verschiedene Mechanismen authentifizieren, neben der Authentifizierung über die interne Datenbank steht noch LDAP und RADIUS zur Auswahl.

Zur Realisierung des Chats beinhaltet ILIAS einen eigenen, Java-basierten Chatserver, der den HTTP Chat Stream bereitstellt. Die Kommunikation zwischen der Plattform und dem Chatserver erfolgt über die Datenbank.

Zur Suche kann bei ILIAS optional ein Lucene-Server ([The]) angebunden werden, der eine

Suche auch in hochgeladenen Dokumenten, beispielsweise PDF Dateien, erlaubt.

2.3.4.3 Moodle

Das LCMS moodle ([Ger07], [CF07], [moo]), welches ebenfalls als Basis PHP und eine relationale Datenbank (MySQL oder Postgres) benutzt, ist in Abbildung 2.3 dargestellt. Die einzelnen Arbeitsmaterialien werden in Kursen organisiert. Die Kurse, welche in hierarchisch organisierten Bereichen angeordnet werden können, bieten zahlreiche Einstellungen wie verschiedene Ansichten, verschiedene Möglichkeiten zur Nutzung von Gruppen sowie verschiedene Anmeldeoptionen.

Deutsch (de) Sie sind angemeldet als Alexander Fleischer (Logout)

moodle Ilmenau
moodle Lernplattform

Willkommen auf der Lernplattform moodle des UnIRZ der TU Ilmenau.

Meine Kurse

- Konstruktionstechnik
- Seminar "Lernplattform für die TU Ilmenau"
- seichte Unterhaltung
- moodle_intern
- Testkurs (fleischer)
- Alle Kurse ...

Mitteilungen

Es liegen keine neuen Mitteilungen vor Mitteilungen...

moodle - news

Supportanfragen / Fehlermeldungen / Neuen Kursbereich einrichten :
Bitte senden Sie eine E-Mail an: imsadmin@tu-ilmenau.de

Das moodle-Team der TU Ilmenau

Kursbereiche

Biomedizinische Technik und Informatik	1
Elektrotechnik	1
Empirische Medienforschung / Politische Kommunikation	14
Kommunikationswissenschaft	2
Maschinenbau	9
Medienkonzeption / Medienpsychologie	1
Medienmanagement	1
Kinderuni 2008	1
Medienforum Ilmenau 2008	2
Medienproduktion	7
Medienwissenschaft	8
Multimediale Anwendungssysteme	2
Technik- und Wirtschaftsgeschichte	5
Technische Informatik und Ingenieurinformatik	7
Verschiedenes	1
Interner Arbeitsbereich	

Hauptmenü

- Kurzanleitung: Teilnehmer
- Kurzanleitung: Dozenten/Kurverwalter
- Dokumentationen
- Fragen & Support
- Hilfe bei "Passwort oder Anmeldenamen vergessen"

Kalender

February 2008

Mo	Di	Mi	Do	Fr	Sa	So
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29		

Online-Aktivitäten

(in den letzten 2 Minuten)

- Alexander Fleischer
- Saskia Polser

Abbildung 2.3: LCMS moodle

Ähnlich wie bei Portalsystemen können bei moodle verschiedene, konfigurierbare Blöcke zusätzlich zum eigentlichen Inhalt definiert werden, welche dann auf der jeweiligen Seite mit angezeigt werden.

Das Basissystem erlaubt das Anlegen und Verwalten von Arbeitsmaterialien als Text, HTML Seite oder hochgeladene Dateien, das Management von Kursterminen des Kurses mit Hilfe eines zentralen Kalenders, das Anlegen von Kursen und Gruppen sowie die Benutzerverwaltung und Administration.

Für die Inhalte selbst verwendet moodle ein flexibles System von verschiedenen Modulen. In der Basisinstallation sind unter anderem folgende Module enthalten:

- Lernmodule

- Lektion

Lektionen ermöglichen die Erstellung von Lernmaterialien. Das Lernmaterial kann Online erstellt werden, und besteht aus Einzelseiten mit eingebundenen Navigationsstrukturen und Multiple Choice Fragen. Es werden 2 Typen von Seiten unterstützt: Verzweigungsseiten und Frageseiten. Abhängig von der ausgewählten Antwort bzw. des Links ist es möglich, jede andere Seite des Moduls anzuspringen. Hierdurch ist es möglich, sehr komplexe Strukturen zu erstellen.

- Arbeitsunterlage

Als Arbeitsunterlage können in moodle Dateien, ganze Verzeichnisse, externe Links und einzelne Seiten, welche auch direkt online erstellt werden können, bereitgestellt werden.

- Aufgabe

Mit Hilfe dieses Moduls können Aufgaben gestellt werden. Es ist dabei auch möglich, einen Abgabetermin festzulegen und die Annahme von Lösungen auf diesen zu beschränken. Lösungen können von Studenten in Form einer Datei oder aber direkt Online eingegeben werden. Zur Bewertung der Aufgaben steht ein übersichtlicher Dialog für Kurs-Tutoren zur Verfügung.

- Datenbank

Mit Hilfe des Datenbankmoduls lassen sich einfach tabellarische Daten erfassen.

Bei Definition der Tabelle werden die einzelnen Felder benannt und der Typ der dieser Felder festgelegt (Datum, Text, Auswahl, etc.). Für die Anzeige der Daten sind entsprechende Templates vorhanden, welche direkt im Online-Editor angepasst werden können, auch die Ausgabe als RSS-Feed ist möglich. Auch ist es über dieses Modul möglich, eine Dateiuploadmöglichkeit für Studenten zur Verfügung zu stellen.

- Glossar

Das Glossar ermöglicht eine alphabetische Liste von Begriffen und deren Bedeutung anzulegen. Dabei können diese nach benutzerdefinierten Kategorien gruppiert werden. Zu jedem Eintrag ist es möglich, eine Datei zur näheren Beschreibung anzuhängen.

- Test

Ein Test kann aus Fragen verschiedenen Typs, welche nach verschiedenen Kategorien eingeordnet sind, zusammengestellt werden. Neben Multiple Choice, Texteingabe, Lückentext und Zuordnungsfragen sind auch Fragen möglich, bei welchen das Ergebnis mit Hilfe einer Formel berechnet wird, so dass es möglich ist, jedem Testteilnehmer „individuelle“ Fragen zu stellen. Dabei können in der Aufgabendefinition Variablen eingesetzt werden, für welche ein Wertebereich vorzugeben ist. Anhand einer eingegeben Formel berechnet moodle das korrekte Ergebnis und prüft die Eingabe entsprechend des erzeugten Wertes. Auch eine Definition und Umrechnung verschiedener Einheiten ist möglich.

- Workshop

Mit Hilfe des Workshopmoduls ist es möglich, eine Aufgabenstellung vorzugeben, welche innerhalb eines festgelegten Zeitraumes von den Teilnehmern zu bearbeiten und abzugeben ist. Die abgegebenen Antworten (auch als Datei möglich) können im Anschluss dann von allen Teilnehmern angeschaut und anhand vorgegebener Kriterien bewertet werden.

- Kommunikationsmodule

- Chat

Moodle benutzt ein auf dem HTTP-Protokoll und JavaScript basiertes Chatsys-

tem. Im Chat ist es möglich, HTML-Code einzufügen. Somit ist auch eine Einbindung von Audio- oder Video möglich. Andere aktive Inhalte wie Java-Applets oder JavaScript werden dagegen gefiltert.

- Abstimmung

Mit Hilfe des Moduls „Abstimmung“ können einfache Umfragen (Single Choice) realisiert werden. Dabei ist es leider nicht möglich, die Ergebnisse der Abstimmung immer zu anonymisieren: Während die anonyme Anzeige des Ergebnisses bei Teilnehmern möglich ist, bekommt ein Kursverwalter immer angezeigt, welcher Teilnehmer wie abgestimmt hat.

- Forum

Moodle bietet ein Forum, bei welchem die Antworten hierarchisch angezeigt werden. Damit ist es einfach möglich herauszufinden, auf welches Posting sich eine Antwort bezieht.

- Wiki

Das Wiki-System erlaubt ein kollaboratives Erstellen von Dokumenten. Durch einen entsprechenden Filter werden existierende Wiki-Seiten auch innerhalb aller anderen Stellen des Kurses verlinkt, so dass ein einfacher Zugriff auf diese Inhalte gewährleistet ist. Die Inhalte werden auf einfache Art mit Hilfe eines Rich Text Editors eingegeben. Daher ist es nicht erforderlich, eine spezielle Syntax für die Formatierung und Verlinkung, wie bei anderen Wiki-Systemen gebräuchlich, zu erlernen.

Moodle bietet eine hohe Integration der Autorenwerkzeuge, alle Materialien lassen sich direkt online bearbeiten. Durch den eingesetzten Rich Text Editor ist es einfach, Formatierungen, Tabellen und Bilder direkt im Browser einzugeben. Dieser verhindert aber auch das direkte Einbinden von „unbekannten“ Elementen wie Multimediainhalten und Java Applets. Die Erstellung einer Navigationsstruktur innerhalb einer Lektion ist etwas unübersichtlich, da es keine übersichtliche Methode gibt, anzeigen zu lassen, wie die Seiten untereinander verbunden sind. Abgesehen von einer Beschreibung ist es nicht möglich, Metadaten zu Inhalten zu erfassen. Eine Suche ist nur innerhalb von Foren möglich.

Moodle kann die Inhalte zum Datenaustausch in ein eigenes XML-Format exportieren. Dabei

kann der Umfang des Exports ausgewählt werden (neben Lerninhalten können auch Forenbeiträge, Benutzerdaten, Logs und viele weitere Daten dort gesichert werden). Ein Export der Inhalte in SCORM oder in andere standardisierte Formate ist nicht vorgesehen.

Technische Realisierung moodle verwendet zwar für die Hauptseite und einige weitere Elemente Templates, die eine Veränderung der Oberfläche zulassen, setzt diese aber nicht so konsequent ein wie beispielsweise ILIAS. In den moodle gibt es daher in vielen Scripten keine Trennung zwischen Verarbeitung und Präsentation, da direkt der entsprechende HTML-Code ausgegeben wird und diese Ausgaben von der Verarbeitungsreihenfolge abhängig sind.

Moodle kennt seit Version 1.8 eine flexible Rollenstruktur zur Festlegung der Rechte der einzelnen Benutzer, die eine detaillierte Konfiguration ermöglichen. Für die 4 festen globalen Rollen, die bis zur Version 1.8 existierten (Gast, Mitglied, Kursverwalter und Administratoren) wurden entsprechende Voreinstellungen angelegt. Die Rollen können flexibel auf jeder Hierarchieebene der Kursstruktur zugewiesen werden, auch für einzelne Kurse sind verschiedene Rollen und damit Rechtezuweisungen für Trainer und Studenten möglich.

Moodle verwendet intern wenige umfangreiche Bibliotheksdateien, welche die Basisfunktionalität des Systems in Form vieler einzelner Funktionen bereitstellen. Diese sind nicht objektorientiert realisiert, aber ausführlich im Quelltext und auf der Webseite dokumentiert.

Moodle lädt beim Aufruf automatisch bestimmte Dateien in einigen Unterverzeichnissen, welche eine Ergänzung von moodle um eigene Plugins sehr einfach gestaltet. Die Plugin-Schnittstelle ist sehr flexibel und gut dokumentiert, es lassen sich nicht nur neue Module als Lerninhalte oder Aktivitätsmodule einrichten, sondern beispielsweise auch eigene Textfilter, welche systemweit für alle Module gelten oder eigene Authentifizierungsmethoden. Beim Laden neuer oder aktualisierter Plugins wird automatisch beim ersten Aufruf der Administration ein evtl. nötiges Update der Datenbank über eine entsprechende API des Plugins durchgeführt.

Dieser Mechanismus wirkt allerdings nur dort, wo dies von moodle vorgesehen ist, bei einigen Funktionen ist es daher z.Zt. noch unmöglich, mit der Hilfe von Plugins das Verhalten anzupassen. Mit der für Ende 2007 angekündigten moodle-Version 1.9 ist allerdings ein neues Ereignissystem angekündigt, welches dies beheben könnte.

2.3.4.4 DotLRN

DotLRN (auch .LRN, [dot]) ist ein Open Source E-Learning System, welches auf dem Community-System OpenACS basiert. dotLRN benötigt als Umgebung den AOL-Server und Postgres, und ist selbst in Tcl geschrieben. Aufgrund des Ursprungs in einem Community-System sind die Kommunikations- und Kollaborationsfunktionen sehr umfangreich (Weblog, Umfragen, News, Forum, E-Mail, FAQ, Dateiablage).

2.3.4.5 Vergleich

Ein Vergleich der vorgestellten LCMS-Systeme ist in Tabelle 2.1 dargestellt. Dabei wurden die in den vorgehenden Abschnitten ausführlich dargestellten Eigenschaften kategorisiert und teilweise für eine übersichtlichere Darstellung zusammengefasst.

2.4 Protokolle zum Datenaustausch

Im folgenden sollen einige Protokolle zum automatisierten Datenaustausch über das Internet betrachtet werden.

2.4.1 Common Object Request Broker Architecture (CORBA)

CORBA ([HJS01], [Obja]) ist eine objektorientierte Middleware, welche es mittels eigener Protokolle wie der Interface Definition Language (IDL) erlaubt, verteilte Anwendungen zu erstellen. CORBA ist dabei unabhängig von der Programmiersprache und Plattform. Es erlaubt den Zugriff auf Objekte unabhängig von deren Standort, vermittelt also transparent den Zugriff auf entfernte Objekte. Durch weitere Protokolle wie dem Dynamic Invocation Interface (DII) ist es auch möglich, auf zur Erstellungszeit unbekannte Services zuzugreifen. Die Verwaltung der Objekte wird komplett von CORBA gekapselt, CORBA unterstützt dabei viele Eigenschaften objektorientierter Programme wie Kapselung, Vererbung und Polymorphismus.

Tabelle 2.1: Vergleich der LCMS Systeme

System	metacoön	ILIAS	moodle	.LRN
Version	3.8	3.8	1.7	2.3
Funktionen - Lernmodule				
Lerninhalte	intern, SCORM, HTML	intern, SCORM, AICC, HTML	intern, HTML, SCORM, AICC	intern, IMS, SCORM
Tests	QTI (teilweise)	intern, QTI (teilweise)	intern, QTI, weitere	intern, QTI
Aufgaben	x	x	x	x
Glossar	x	x	x	x
Links	x	x, automatische Prüfung	x	x
Funktionen - Kommunikation				
Mail	nur extern	intern, extern	intern, extern	intern, extern
Forum	x	x	x	x
Chat	?	HTTP	HTTP	HTTP
Umfrage	x	x	x	x
Wiki	-	-	x	-
Lerngruppen	-	x	x	x
weitere	-	-	Weblog	Weblog
technische Realisierung				
Basissystem	PHP, MySQL, Java, Co- coon	PHP, MySQL (oder andere Datenbank), Java, Lucene	PHP, MySQL	TCL, AOLServer
Authentifikation	intern	intern, LDAP, Radius	intern, LDAP, Radius, IMAP, weitere	intern, LDAP, Radius, wei- tere
Autorisierung	Unix-ähnlich	globale und lokale Rollen	globale Rollen	globale Rollen

2.4.2 Remote Procedure Call (RPC)

RPC ist ein Protokoll, welches den Aufruf von Funktionen auf entfernten Systemen ermöglicht. Die erste Version des Protokolls wurde 1988 als RFC standardisiert. Die heute verwendete Version 2 des Protokolls ist in RFC 1057 [Sun88] und RFC 1831 [Sri95] beschrieben. RPC ist dabei plattformspezifisch, verschiedene Implementierungen verwenden andere Übertragungsprotokolle und sind nicht austauschbar. Um RPC zu nutzen, werden aus einer Schnittstellenbeschreibung (Remote Procedure Call Language) so genannte Stubs (Funktionsrümpfe) für Client und Server erzeugt. Auf Serverseite sind die entsprechenden Funktionen zu implementieren, welche auf Seite des Clients dann durch einen entsprechenden Stub zur Verfügung gestellt werden, so dass ein Aufruf auf Clientseite direkt (als wäre es eine lokale Funktion) erfolgen kann. Die Aufrufe sind dabei immer synchron, während eines Aufrufs einer RPC Funktion ist der Client blockiert, bis er eine entsprechende Antwort vom Server erhält.

2.4.3 XML-basierte Webservices

XML-basierte Webservices nutzen zur Übertragung der Daten das Hypertext Transfer Protocol (HTTP, [FGM⁺99]). Die Inhalte werden als XML ([LFL06a]) übertragen. Diese Art der Übertragung und Codierung des Inhaltes bietet zahlreiche Vorteile:

- HTTP Server gibt es für zahlreiche Plattformen, Betriebssysteme und mit Schnittstellen zu zahlreichen Programmiersprachen für die Anbindung externen Module.
- Die Realisierung der Funktionalität kann unabhängig vom Server erfolgen, da HTTP Server das Einbinden externer Module ermöglichen.
- Die Übertragung der Daten im XML Format garantiert die Lesbarkeit auf allen Systemen, unabhängig von Plattform oder Programmiersprache. Gleichzeitig wird ein einfacher Zugriff auf die Daten gewährleistet und eine einfache Umwandlung in andere Formate, z.B. per XSLT ([LFL06b]), ermöglicht.
- Durch XML-Schemata kann das Datenformat detailliert definiert und beschrieben werden, was die Entwicklung von Anwendungen, welche die Daten verwenden, vereinfacht.

Aufgrund des zustandlosen Aufbaus von HTTP ist zuerst nur eine einfache Anfrage → Antwort Struktur möglich, jede Anfrage ist eine für sich abgeschlossene Einheit, die unabhängig von evtl. vorher getätigten Anfragen ist. Dies lässt sich aber durch die Aufnahme entsprechender Zustandsinformationen wie z.B. Session-IDs innerhalb der übertragenen XML-Daten umgehen. Weiterhin sind bei Webservices die Rollen von Client und Server fest definiert, so dass z.B. eine Realisierung von Callback-Mechanismen nicht möglich ist.

Die im folgenden genannten Protokolle definieren jeweils ein XML-Schema, welches das übertragene Datenformat beschreibt. Zusätzlich gibt es bei vielen der genannten Protokolle Implementierungen von Bibliotheken in verschiedenen Programmiersprachen, welche die Nutzung dieses Services erheblich vereinfachen.

XML-RPC XML-RPC ist die Weiterentwicklung des in Abschnitt 2.4.2 beschriebenen RPC-Protokolls. Auch bei XML-RPC werden nur synchrone Funktionsaufrufe unterstützt, die Standardisierung des Übertragungsprotokolls ermöglicht aber den Austausch zwischen verschiedenen Implementierungen. Das XML Schema definiert nur relativ einfache Strukturen zum Aufruf von Funktionen, zur Übergabe von Parametern (wobei neben einfachen Datentypen auch Strukturen und Felder übergeben werden können) und zur Rückgabe von Ergebnissen. Welche Parameter dabei in welcher Reihenfolge übergeben werden und wie eine entsprechende Antwort aussieht, wird von der entsprechenden Funktion auf dem Server bestimmt. Es gibt keine zentrale Beschreibung der Schnittstellen der einzelnen Funktionen des Servers.

Simple Object Access Protocol (SOAP) SOAP ([STK02], [WCL⁺05]) wurde ursprünglich von Dave Winer und Microsoft entwickelt, und erstmalig 1999 veröffentlicht. Später wurde SOAP (Version 1.1) als Standard beim World Wide Web Consortium (W3C) eingereicht ([BEK⁺]). Dabei wurde das Ziel verfolgt, eine Arbeitsgruppe für die Weiterentwicklung von SOAP zu gründen.

Eine SOAP Nachricht besteht aus einem Header, welcher zum Beispiel Metadaten, Authentifizierungsinformationen oder Angaben zur Verschlüsselung über das Objekt beinhalten kann, und einem Body mit den Nutzdaten der Nachricht. Den Aufbau der (XML-)Nutzdaten bestimmt allein der angesprochene Service. Dieser Aufbau einer Nachricht kann sich durch den

relativ großen Overhead und den Aufwand beim Parsen der XML-Datei auch als Nachteil erweisen.

SOAP kann verschiedene Transportprotokolle für die Übertragung der Nachrichten verwenden, am häufigsten wird jedoch HTTP eingesetzt.

Auf SOAP basierende Webservices werden häufig mit einer WSDL (Web Service Description Language) Datei beschrieben. Diese XML-Datei enthält Informationen zu den angebotenen Funktionen, den erwarteten Parametern, den Rückgabewerten und dem Pfad zum Aufruf der Funktion. Neben den Standarddatentypen aus XMLSchema können auch z.B. eigene komplexe Datentypen definiert werden, es kann angegeben werden, wie oft ein Element auftreten kann oder muß und es können Regeln für die Validierung der Daten angegeben werden [CCMW]. Mit Hilfe dieser Beschreibung ist es sehr einfach, eine entsprechende Clientanwendung zu erstellen, da die Schnittstellenbeschreibung mit WSDL in maschinenlesbarer Form vorliegt und von entsprechenden Bibliotheken direkt verwendet werden kann.

Für SOAP und WSDL gibt es Bibliotheken für viele Programmiersprachen, welche es ermöglichen, aus einer vorhandenen Klassenstruktur relativ einfach eine SOAP-Schnittstelle zu erzeugen (Server) oder aber bei Vorliegen der WSDL-Beschreibung das SOAP-Objekt direkt in einer Anwendung nutzbar machen.

Mit UDDI (Universal Description, Discovery and Integration) gibt es bereits ein auf SOAP basierendes Protokoll für einen Verzeichnisdienst für Web Services. Verschiedene Anbieter stellen einen entsprechenden Dienst zur Verfügung [CHRR].

Content Syndication Protokolle - RSS, Atom, ICE Content Syndication dient dem automatischen Austausch von Inhalten (Nutzdaten) wie Nachrichten und Artikeln zwischen verschiedenen Anwendungen. Die dabei zur Datenübertragung eingesetzten Protokolle basieren zum größten Teil auf XML.

RSS RSS (Really Simple Syndication) ist ein sehr einfaches Content Syndication Protokoll. RSS spezifiziert ein auf XML basierendes Format, welche Titel, Kurzbeschreibung und Links zu mehreren Artikeln enthalten kann. Die Elemente sind dabei fest vorgegeben, eigene

Strukturen können nicht übertragen werden ([RSS]).

Die Übertragung der Daten ist dabei nicht Bestandteil der Spezifikation. Meist wird dieses Format dynamisch vom Server erzeugt und kann mittels eines HTTP GET Request (welcher ggfs. auch Parameter zur näheren Spezifizierung der angefragten Inhalte enthalten kann) abgerufen werden.

Atom Atom ist ein Content Syndication Protokoll, welches in [NS05] standardisiert wurde. Es enthält neben dem eigentlichen Inhalt im Text-, HTML- oder XHTML-Format Metadaten wie Autor und Zusammenfassung. Analog zu RSS definiert Atom nur den Aufbau und die Elemente der XML-Datei.

Information and Content Exchange (ICE) Das ICE Protokoll [WOH⁺], [BCK⁺], welches 1998 von mehreren Inhaltsanbietern und Softwareentwicklern entwickelt wurde, stellt eine auf XML basierende Content Syndication Schnittstelle bereit. Anders als die oben genannten Lösungen RSS und Atom enthält ICE zahlreiche weitere Funktionen zur Verwaltung des Datenaustausches. So ist es beispielsweise möglich, sich verfügbare Kanäle anzeigen zu lassen und Kanäle zu abonnieren oder das Abonnement zu ändern. Im Protokoll werden sowohl Authentifizierungs- und Abrechnungsfunktionen, als auch die Übermittlung der gewährten Verwertungsrechte unterstützt.

ICE benutzt eine Client-Server Architektur, um die Daten zur Verfügung zu stellen. Bei der Verwendung von Callbacks zur Ankündigung neuer Nachrichten wird eine Server-Server Architektur vorausgesetzt. ICE in der Version 1.x Überträgt alle Daten mit Hilfe von XML-Dokumenten, welche über HTTP-POST übertragen werden. Die aktuelle Version 2.0 von ICE benutzt dagegen zur Übertragung SOAP.

2.5 Standards zum Austausch von Lehrinhalten

Lehrinhalte liegen meist in verschiedenen, plattformspezifischen Formaten vor. Um einen Austausch zwischen verschiedenen Plattformen zu ermöglichen, wurden verschiedene standardisierte Formate geschaffen, welche den Austausch von Lernmaterialien auch zwischen

verschiedenen Plattformen, Architekturen und Systemen ermöglichen sollen ([Pai06]).

2.5.1 LOM

LOM steht für Learning Objects Metadata und ist ein Standard, der die Suche, die Nutzung und den Austausch von Lernobjekten erleichtern soll ([LOM02], [Kla02]). Dazu werden zu jedem Lernobjekt verschiedene Metadaten zugeordnet. Diese sind in folgende Hauptkategorien aufgeteilt:

1. General Category (Allgemein): Grundlegende Informationen, die das Lernobjekt als Ganzes beschreiben.
2. Lifecycle Category (Lebenszyklus): Eigenschaften, die einerseits die Geschichte und den aktuellen Zustand des Lernobjektes als auch die beeinflussenden Lernobjekte beschreiben.
3. Meta-Metadata Category (Metametadaten): Merkmale der Metadatenbeschreibung an sich.
4. Technical Category (Technische Details): technische Voraussetzungen und Merkmale des Lernobjekts.
5. Educational Category (Pädagogische Details): Bildungsmerkmale und pädagogische Beschreibung des Lernobjekts.
6. Rights Category (Rechte): Über Nutzungsbedingungen des Lernobjekts und Copyright-Fragen wird informiert.
7. Relation Category (Verwandte Ressourcen): Beziehungen zwischen dem Lernobjekt und anderen verwandten Lernobjekten.
8. Annotation Category (Anmerkung): ermöglicht Anmerkung über den Bildungsnutzen des Lernobjekts und Informationen über die Entstehung der Kommentare (wann, von wem)
9. Classification Category (Klassifikation): Einordnung des Lernobjekts in ein Klassifizierungssystem.

Unterhalb dieser Hauptkategorien gibt es zahlreiche, auch verschachtelte Unterobjekte, welche die eigentlichen Metadaten enthalten. Textinformationen können dabei jeweils in mehreren Sprachen erfasst und gespeichert werden, für andere Felder gibt es definierte Wertebereiche, was eine Filterung nach diesen Kriterien vereinfacht. Die Daten werden als XML-Beschreibung abgelegt und werden auch in anderen XML-basierten Formaten, wie z.B. SCORM, verwendet.

2.5.2 Dublin Core

Dublin Core ist ein weiteres Format zur Spezifikation von Metadaten für Dokumente. Dublin Core ist dabei nicht auf Lernobjekte beschränkt, sondern wird beispielsweise auch eingesetzt, um Webseiten zu beschreiben. XML ist dabei nur eines von mehreren Formaten zur Angabe dieser Metadaten. Dublin Core definiert unter anderem die Felder Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage und Rights, wobei es keine genaue Festlegung über den Inhalt der einzelnen Felder gibt.

2.5.3 Aviation Industry CBT Committee (AICC)

AICC ist ein von der amerikanischen Luftfahrtindustrie erschaffenes Gremium, welches Richtlinien für E-Learning Angebote entwickelt. Unter anderem wurden auch Spezifikationen für den Austausch von Lehrinhalten entwickelt. Zum Austausch von Lehrinhalten wird dabei ein Archiv benutzt, in welchem sich, neben den Lehrinhalten in der Form von HTML-Seiten, einige weitere Dateien im CSV oder INI-Format mit Metadaten zum jeweiligen Lehrinhalt, Informationen zu den beinhalteten Blöcken und Elementen, die Navigationsstruktur und den Voraussetzungen für den Aufruf von einzelnen Blöcken, befinden.

2.5.4 SCORM

SCORM (Sharable Content Object Reference Model) ist ein Satz von Standards für LMS. Ziel von SCORM ist es, einen allumfassenden Standard für die Interoperabilität, die Zugriffsmöglichkeit und die Wiederverwendbarkeit von Lernobjekten zu schaffen.

SCORM ist in einem Satz von Büchern ([ADL]) beschrieben. Diese „Sammlung“ von „technischen Büchern“ besteht aus vier Bestandteilen:

- The SCORM 2004 Overview Book – beschreibt Geschichte und Zweck der ADL Initiative und SCORM inklusive der Spezifikationen und Standards, die SCORM verwendet. Es zeigt auch, wie die verschiedenen Bücher miteinander verbunden sind.
- The SCORM Content Aggregation Model (CAM) Book – beschreibt, wie man Lerninhalte für Wiederverwendbarkeit und Austausch zwischen Systemen zusammenpacken kann. Dieses Modell enthält folgende Komponenten:
 - Content Model – beschreibt allgemein die SCORM CAM Konzepte
 - Content Packaging – beschreibt, wie SCORM Content Packages aufgebaut sind, wie man Lerninhalte damit strukturieren und austauschen kann. Das Content Package besteht aus einer oder mehreren Manifest-Dateien (XML-Dokumente), welche die Daten des Content Package beschreiben, und den eigentlichen Daten
 - Metadaten (Meta-data) – beschreiben Lerneinheiten nach LOM-Standard
 - Sequencing und Navigation – definiert einen Satz von Regeln, der die Reihenfolge und Navigation innerhalb von SCORM Lerneinheiten beschreibt
- The SCORM Run-Time Environment (RTE) Book – beschreibt Anforderungen an Learning Management Systeme (LMS) für Verwaltung von SCORM Lerneinheiten. Es besteht aus drei Komponenten:
 - Runtime Environment Management: Starten von Lernobjekten (launching of content objects) – SCO (sharable content object) und Assets, Verwaltung der Kommunikation mit SCO, Verwaltung des Runtime Environment Data Model.
 - Application Programming Interface (API): definiert einen Satz von Funktionen, welche eine Schnittstelle zwischen LMS und SCO bilden. Diese Schnittstelle ermöglicht die Kommunikation zwischen LMS und SCO.
 - Runtime Environment Data Model: stellt ein Vokabular zur Verfügung, das benutzt wird, um Information zwischen LMS und SCO unter Verwendung der SCORM API Funktionen („get“, „set“) auszutauschen.

- The SCORM Sequencing and Navigation (SN) Book – beschreibt den Ablauf und die Navigation innerhalb von Lernobjekten. Die Inhalte werden mit Hilfe von Navigationsereignissen, die von Lernenden oder von Lernsystemen ausgelöst werden können, geordnet. Verzweigungen und Datenflüsse können mit einer vordefinierten Menge von Aktivitäten beschrieben werden, welche im Allgemeinen bei Entwurf des Lernobjekts festgelegt werden.

SCORM Lerneinheiten setzen sich aus Shareable Content Objects (SCOs) zusammen, welche jeweils kleine, wiederverwendbare Objekte darstellen. Diese benutzen die SCORM API, um auf Funktionen des LCMS zuzugreifen, beispielsweise um Werte wie Lernfortschritt und Testergebnisse zu speichern oder abzurufen. Diese SCOs können mit der Hilfe von XML-Dateien zu Lerneinheiten verbunden werden, dabei ist eine detaillierte Angabe der Navigationsstruktur, den Voraussetzungen, etc. möglich.

SCORM bietet leider keine semantische Definition der Inhalte, SCO's beinhalten immer (beliebige) HTML-Dateien mit eingebetteten Scripten, welche das JavaScript-API nutzen. Eine semantische Definition der Inhalte (Definitionen, Beispiele, Aufgaben,, etc.), beispielsweise über eine Auswahl von vorgegebenen CSS Klassen, um diese für das LCMS auch für Recherchefunktionen besser zu erschließen, ist mit SCORM nicht möglich, das LCMS muss sich hier auf die Metadaten verlassen.

2.5.5 IMS Question & Test Interoperability (IMS QTI)

IMS Question & Test Interoperability [IMS] ist ein vom IMS Global Learning Consortium definiertes Datenformat für den Austausch von Online-Tests, Quizzes und Umfragen.

Das auf XML basierende Datenformat ist sehr umfangreich und beinhaltet unter anderem:

- Die Zusammenfassung und Strukturierung von Fragen zu verschiedenen Abschnitten und Blöcken
- die Definition von verschiedenen Fragetypen, wie Mehrfachauswahl, Einfachauswahl, Zuordnung, Lückentext, Upload und weitere Fragetypen

- die Einbindung von zusätzlichem Material und Erläuterungen zu einzelnen Fragen
- die flexible Festlegung von Bewertungskriterien mit Hilfe einer komplexen, in XML definierten Logik

2.5.6 Weitere Formate

Für den Austausch von Lerneinheiten zwischen gleichartigen Systemen wird von den meisten LCMS ein für dieses LCMS spezifisches Exportformat unterstützt. Diese LCMS spezifischen Formate sind jedoch nur für den Austausch zwischen gleichartigen Plattformen geeignet, teilweise gibt es auch Inkompatibilitäten zwischen verschiedenen Versionen einer Plattform.

2.6 Technologien für die Präsentation interaktiver Lehrinhalte

In diesen Abschnitt werden die verbreitetsten Techniken zur Darstellung interaktiver Inhalte innerhalb einer Webseite betrachtet.

2.6.1 Java

Java ist eine plattformunabhängige Programmiersprache ([Sch06], [Sun]). Java-Programme werden in Bytecode übersetzt, welcher auf dem jeweiligen Zielsystem mit Hilfe einer virtuellen Maschine ausgeführt wird. Zielsysteme sind dabei nicht nur PC's, virtuelle Maschinen für Java sind auch für viele mobile Endgeräte verfügbar.

Java ist universell einsetzbar, es gibt zahlreiche Bibliotheken, welche das Lösen von Standardproblemen vereinfachen. Allerdings müssen einige Komponenten, wie zum Beispiel das Java Media Framework für die Wiedergabe von Audio- und Videodateien, zusätzlich beim Anwender nachinstalliert werden, um die entsprechenden Bibliotheken einsetzen zu können.

Java ist durch die Verwendung eines Sicherheitsmanagers, welcher bei in Webseiten eingebetteten Programmen (Applets) alle Zugriffe überwacht, sehr sicher.

Der Anwender muss bei der Verwendung von Java-Applets eine entsprechende virtuelle Ma-

schine für sein System installiert haben. Viele Anwender müssen diese erst nachinstallieren, was über langsame Verbindungen aufgrund der Größe des Downloads problematisch ist.

2.6.2 Adobe Flash

Das von Macromedia entwickelte Flash-Format wird hauptsächlich verwendet, um Animationen in Webseiten zu erstellen. Dabei ist Flash in der aktuellen Version jedoch nicht mehr auf reine Animationen beschränkt, durch die Verwendung der eingebauten Scriptsprache ActionScript sind auch komplexere Simulationen möglich.

Zum Abspielen dieser Animationen wird auf der Seite des Anwenders ein Plugin für den Webbrowser benötigt, welches für verschiedene Plattformen kostenlos erhältlich ist. Die Installation des Plugins durch den Anwender ist sehr einfach, auch die Größe des Downloads ist im Vergleich zu Java geringer.

Neben einer kommerziellen Autorenumgebung ([Ado], [Wes07]) von Adobe gibt es einige freie Bibliotheken, welche das Erstellen von Flash-Filmen ermöglichen, allerdings nicht den kompletten Funktionsumfang und Komfort der kommerziellen Autorenumgebung bieten.

2.6.3 JavaScript / AJAX

JavaScript ist eine Programmiersprache, welche in den meisten Browsern integriert ist ([Koc06]). Das hat den Vorteil, dass es auf vielen Systemen direkt und ohne die Installation eines zusätzlichen Plugins verfügbar ist. In aktuellen Browsern bietet Javascript umfangreiche Möglichkeiten die Struktur und die Darstellung eines Dokumentes mit Hilfe des Document Object Models zu manipulieren und auf Eingaben des Anwenders durch eine entsprechende Ereignisbehandlung zu reagieren. Damit sind auch komplexere Animationen oder die Simulation und Darstellung komplexerer Zusammenhänge realisierbar.

JavaScript ist allerdings nicht vollständig unabhängig vom verwendeten Browser. Während ein standardisierter Mindestsprachumfang mit allen Browsern funktioniert, sind viele andere Funktionen abhängig vom verwendeten Browser und müssen für verschiedene Browser in anderer Art und Weise realisiert werden.

Einer der größten Nachteile von JavaScript ist das Fehlen eines Objektes, welches eine Zeichenfläche zur Verfügung stellt. So ist es mit JavaScript zwar möglich, bestehende Grafiken nachzuladen, gegen andere auszutauschen und zu verschieben, aber das programmgestützte Zeichnen von Elementen ist nicht möglich, so dass sich grafisch interaktive Elemente damit nicht realisieren lassen.

Es ist mit Hilfe des XMLHttpRequest Objektes möglich, Daten im Hintergrund, unabhängig von der angezeigten Webseite, zu übertragen, so dass eine Kommunikation mit dem Server möglich ist, ohne die entsprechende Seite neu laden zu müssen. Diese Kombination von JavaScript, asynchroner Übertragung und XML als Format für die Datenübertragung wird unter dem Acronym AJAX (*Asynchronous JavaScript and XML*) zusammengefasst.

Für AJAX existieren eine Reihe von Bibliotheken und Frameworks, welche eine Abstraktion auf eine browserunabhängige Schnittstelle bieten und darüber hinaus zahlreiche Objekte für Standardanwendungen wie Dialoge, Eingabeüberprüfungen und natürlich die Hintergrundkommunikation mit dem Server zur Verfügung stellen ([ML06]).

2.7 Online Bezahlungssysteme

Im folgenden Abschnitt werden ausgewählte Bezahlungssysteme beschrieben. Dabei wurden als Beispiele solche Verfahren ausgewählt, bei welchen eine zeitnahe Bestätigung des Bezahlvorganges möglich ist, die Bezahlung also direkt Online abgewickelt wird. Es wurden nur Verfahren betrachtet, welche das Übergeben der Daten direkt per HTTP unterstützen, so dass eine einfache Integration in eLearning Systeme möglich ist. Bei diesen HTTP basierten Bezahlungssystemen läuft der Bezahlvorgang nach folgendem Schema ab (Abbildung 2.4):

- Der Kunde gelangt über einen Link (HTTP GET) oder ein Formular (HTTP POST) auf die Seite des Zahlungsanbieters. Dabei werden alle benötigten Daten, wie Accountdaten des Zahlungsanbieters, Produktinformationen und Preis als Parameter übergeben.
- Der Anbieter führt die Bezahlung durch, beispielsweise durch Abbuchung des Betrages von einer Kreditkarte.
- Bei erfolgreicher Bezahlung wird der Kunde über ein Formular oder einen speziellen

Link wieder zurück zur Ursprungsseite geleitet. Über die in diesem Link enthaltenen Parameter kann das System feststellen, ob die Bezahlung erfolgreich war. Dabei wird meist eine zusätzliche Komponente, wie eine beim Anbieter hinterlegte Kennung oder eine Prüfsumme übertragen, welche es ermöglicht, die Echtheit der Antwort zu prüfen. Alternativ dazu kann das Senden der Bestätigung auch direkt vom Anbieter des Bezahl-systems über eine Server-to-Server Kommunikation über HTTP übertragen werden.

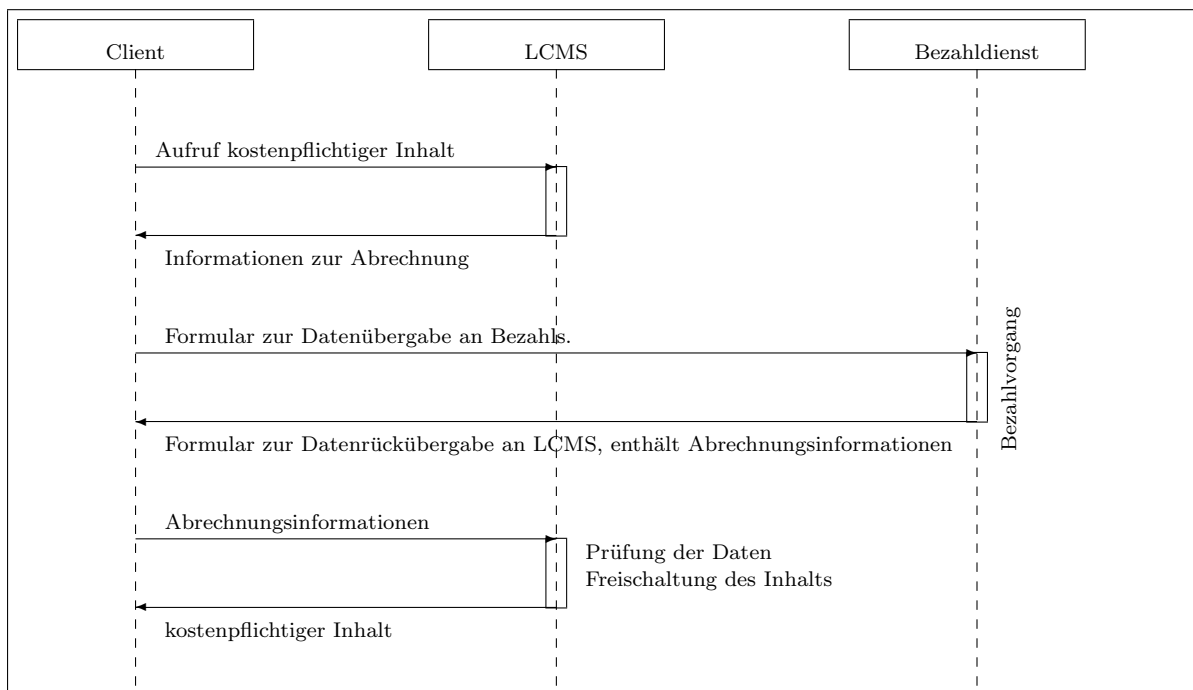


Abbildung 2.4: Ablauf des Bezahlvorgangs bei verbreiteten HTTP-basierten Bezahlssystemen

2.7.1 Paybest

Paybest ([Nüt05], [Nüt02], [4FO]) vereinigt mehrere Bezahlssysteme unter einer Schnittstelle, unter anderem Bezahlung per Anruf, Moneybookers, T-Pay, PayPal, Firstgate und Bezahlung per Überweisung. Mit Ausnahme der letzten Option erfolgt eine sofortige Bestätigung des Kaufes per Aufruf einer entsprechenden URL. Bei der Bezahlung der Überweisung kann, nach Eingang des Geldes, auch eine automatische Bestätigung ausgelöst werden, diese dann jedoch um die übliche Banklaufzeit von 3-5 Tagen verzögert.

Die Übertragung der Daten wird bei Paybest mit Hilfe von MD5 Prüfsummen gesichert, so

das eine Manipulation der Daten ausgeschlossen ist und die Antworten verifiziert werden können.

2.7.2 PayPal

PayPal [Paya], ein Unternehmen von ebay, ermöglicht Zahlungen per Bankkonto (Lastschrift) oder Kreditkarte, und bietet dieses Zahlungssystem international an. Paypal unterstützt mehrere verschiedene Arten der Parameterübergabe. Bei der unverschlüsselten Übergabe der Daten wird die Rückleitungs-URL im Klartext übergeben, was Manipulationen durch den Kunden ermöglicht. Erst die optionale, auf openssl / PKCS7 basierte, verschlüsselte Form der Datenübergabe erlaubt eine sichere Übergabe der Zahlungsdaten. Desweiteren unterstützt Paypal die sichere Übertragung der Daten per SOAP direkt von Server zu Server ohne Beteiligung des Clients [Payb].

2.8 Sicherheit und rechtliche Aspekte

2.8.1 Übertragungssicherheit

Zur Sicherung des Übertragungsweges zwischen Benutzer und LCMS wird häufig SSL/TLS eingesetzt, welches in [DA99], [DR06] standardisiert ist. SSL/TLS bietet dabei nicht nur eine verschlüsselte Übertragung der Inhalte, sondern es ist auch eine Authentifizierung des Kommunikationspartners über Zertifikate möglich ([Res00]).

Nach dem Verbindungsaufbau werden dazu beim Handshake Zertifikate ausgetauscht, anhand derer die Identität des Servers (und optional auch des Clients) festgestellt werden kann. Es wird ein Sitzungsschlüssel ausgehandelt, welcher, asymmetrisch mit dem Zertifikat des Servers verschlüsselt, zwischen beiden Kommunikationspartnern ausgetauscht wird. Alle Daten werden zur Übertragung dann mit Hilfe dieses Sitzungsschlüssels symmetrisch verschlüsselt.

2.8.2 Authentifikation und Autorisierung

Wie im Abschnitt 2.3 bereits bei Vorstellung der einzelnen LCMS betrachtet, werden verschiedene Verfahren eingesetzt, um einen Benutzer eines Systems zu Erkennen (Authentifikation) und dessen Rechte im System festzulegen.

Die verbreitetste Methode zur Authentifikation ist die Vergabe von Benutzernamen und Passwörtern. Dabei weist sich der Benutzer anhand der Kenntniss eines geheimen Merkmals (des Passwortes) gegenüber dem System aus.

Das Passwort selbst wird aus Sicherheitsgründen bei den meisten Systemen nicht im Klartext gespeichert, sondern lediglich das Ergebnis einer Einwegfunktion (Hash), mit welchem dann das vom Benutzer gesendete Passwort ebenfalls verschlüsselt wird, um festzustellen, ob beide gleich sein. Dies hat den Vorteil, dass, selbst wenn die Passwortdatenbank des Systems kompromittiert werden konnte, die eigentlichen Passwörter weiterhin geheim sind, da das Rückrechnen des Hashes auf das ursprüngliche Passwort nicht möglich ist.

Die Passwortdatenbank selbst muss dabei nicht unbedingt von dem Dienst geführt werden, an welchem sich authentifiziert werden soll, ebenso ist es bei vielen Systemen möglich, eine Authentifikation gegen externe Datenbanken, wie beispielsweise RADIUS Server, Verzeichnisdienste oder die Authentifikation des Betriebssystems durchzuführen.

Nachteil dieser Methode ist die Abhängigkeit von nur einem geheimen Merkmal, das einfach weiterzugeben ist. Passwörter können auch nicht beliebig komplex sein, da der Benutzer sich diese auch noch merken können muss. Gegen das Abhören der Übertragung sind auch weitere Absicherungen erforderlich, entweder durch eine zusätzliche Verschlüsselung der Daten auf dem Client oder durch die Sicherung der gesamten Übertragung.

2.8.3 Datenschutzrechtliche Aspekte

Bei Benutzung eines LCMS fallen zahlreiche Daten an, bei denen es sich um personenbezogene Daten handelt. Diese personenbezogenen Daten sind nach dem Bundesdatenschutzgesetz ([bds]) besonders zu schützen. Dies betrifft unter anderem folgende Daten:

- persönlichen Angaben der Nutzer (Name, E-Mail Adresse, etc.),
- Daten zur Abwicklung von Zahlungen,
- persönliche Kommentare und Nachrichten,
- Ergebnisse von Prüfungen und Tests,
- Informationen zum Lernfortschritt,
- Informationen über Zugriffe auf Lehrmaterialien,
- IP Adressen.

Es wird dabei durch das LCMS sichergestellt,

- das nur solche Daten gespeichert werden, welche unbedingt erforderlich sind,
- der Umfang der erhobenen Daten festgelegt und dokumentiert ist (Datenschutzerklärung),
- der Zugriff auf diese Daten nur durch berechtigte Personen erfolgen kann.

Jedes der betrachteten LCMS kann den Zugriff auf diese Daten über entsprechende Einstellungen auf bestimmte Gruppen bzw. Rollen einschränken. ILIAS unterstützt zusätzlich auch an mehreren Stellen die Anonymisierung der Daten. So ist es in ILIAS beispielsweise möglich, Ergebnisse von Tests oder den Lernfortschritt innerhalb eines Kurses anonymisiert auszuwerten.

2.8.4 Urheberechtliche Aspekte

Der Urheber digitaler Lerninhalte genießt, wie jeder Urheber eines Werkes, Rechte nach dem Urhebergesetz. Er darf über die Vergabe von Nutzungsrechten frei entscheiden. Diese Nutzungsrechte umfassen unter anderem die Rechte auf

- Vervielfältigung

- Verbreitung
- öffentliche Zugänglichmachung
- Bearbeitung und Umgestaltung

Zur Wahrnehmung und Durchsetzung dieser Rechte gibt es verschiedene technische Ansätze, welche im folgenden kurz erläutert werden. Diese sind notwendig, da digitale Inhalte jederzeit verlustfrei kopiert werden können, und nur mit Hilfe zusätzlicher technischer Verfahren eine Kontrolle und Ausübung der Rechte durchgesetzt werden kann, sofern dies vom Urheber gefordert wird.

2.8.4.1 Markierung der Inhalte

Eine Möglichkeit, die Rechte durchzusetzen, ist die Markierung der Inhalte mit einem nutzerspezifischen Kennzeichen. Dieses Kennzeichen ist einmalig für jeden legitimen Nutzer und wird als “digitales Wasserzeichen” in die jeweiligen Inhalte eingebettet. Diese Markierung kann zwar die Weitergabe sowie die Herstellung von Kopien der Inhalte nicht verhindern, allerdings ist damit zweifelsfrei nachweisbar, wer diese Inhalte in Umlauf gebracht hat, so dass damit eine Verfolgung des Verstoßes auf rechtlichem Wege ermöglicht wird.

2.8.4.2 Bindung der Inhalte an Hardware

Ein weiteres Verfahren ist die Bindung der Inhalte an Hardware. Dieses wird bei den meisten Digital Rights Management Systemen (DRM) Verfahren eingesetzt.

Diese Systeme arbeiten mit einer Verschlüsselung des Inhaltes. Der Schlüssel, welchen den Zugriff auf den geschützten Inhalt erlaubt und meist als Lizenzdatei getrennt vom eigentlichen Inhalt vorliegt, wird an die Hardware des Rechners (oder anderer Geräte) gebunden, indem beispielsweise Seriennummern von Hardwarekomponenten in die Berechnung des Schlüssels mit einfließen. Somit kann ein entsprechend lizenzierter Inhalt nur auf den Systemen abgespielt werden, für welche eine entsprechende Lizenz ausgestellt wurde, da bei einer Weitergabe der Lizenz aufgrund der anderen Hardware der Inhalt nicht mehr entschlüsselt werden kann. Die

Lizenz enthält neben dem Schlüssel zur Entschlüsselung des Inhalts meist auch Angaben wie erlaubter Nutzungsumfang, Zeitbeschränkungen, etc.

Viele kommerzielle Programme unterstützen DRM, beispielsweise Windows Media, bei welchem sich Audio- und Videodateien per DRM geschützt übertragen lassen. Die Vergabe der Lizenzen erfolgt dabei über einen von mehreren verschiedenen Service Provider, welcher im jeweiligen geschützten Dokument referenziert ist. Auch bei Adobe Acrobat ist es möglich, einen Adobe Policy Server zur Vergabe von Lizenzen für geschützte PDF-Dateien zu verwenden. Damit ist ebenfalls ein entsprechender Schutz der Inhalte und eine zentrale Verwaltung von Rechten möglich.

Allen kommerziellen Lösungen ist gemeinsam, dass man sich mit der Auswahl der jeweiligen DRM-Systems an diesen Anbieter bindet. Neben den Kosten für die Nutzung der Dienste ist auch nur mit den Programmen des entsprechenden Systems ein Zugriff auf die jeweiligen geschützten Inhalte möglich. Bei Ausfall des entsprechenden Services zur Lizenzierung sind die jeweiligen nicht mehr zu entschlüsseln und damit verloren.

Ein Ausweg könnte die Etablierung von Open Source DRM Lösungen wie openIPMP [Objb] sein. Durch den Zugriff auf den Quellcode und der Offenlegung der Verschlüsselungsverfahren ist es hier möglich, unabhängig von einem bestimmten Anbieter ein DRM System zu etablieren.

Kapitel 3

Anforderungen an e-Learning Systeme für den Einsatz im erlösorientierten Bildungsexport

In dem folgendem Abschnitt werden die Anforderungen betrachtet, welche beim Einsatz eines LCMS im erlösorientierten Bildungsexport von hoher Bedeutung sind. Neben speziellen Anforderungen an das LCMS selbst werden hier der Datenaustausch zwischen verschiedenen Plattformen, die Präsentation von Multimediainhalten sowie verschiedene Sicherheitsaspekte erläutert.

Die Anforderungen wurden systematisch anhand der im Bildungsexport wichtigen Aspekte aufgestellt. Neben den allgemeinen Anforderungen an das LCMS ist hier vor allem wichtig, einen sicheren Zugang für die Projektpartner einzurichten, der auch das Anbieten der Inhalte gegen Bezahlung umfasst, und einen Datenaustausch mit den verschiedenen Projektpartnern zu realisieren. Ebenfalls wurden einige Anforderungen an Lehrinhalte betrachtet. Diese spielten eine besondere Rolle im Rahmen des als Beispiel betrachteten Projektes „Deutsche Ingenieur fakultät am MEI“ erstellt, da dort parallel zu den Inhalten auch die Sprache trainiert werden sollte, und dazu auch gesprochene Texte und andere multimediale Inhalte verwendet werden sollten.

Die Anforderungen wurden erweitert und ergänzt, um eine Kompatibilität mit anderen Systemen zu ermöglichen und allgemeine und formale Definitionen der Schnittstellen zu erfassen.

3.1 Anforderungen an das LCMS

In den folgenden Abschnitten werden die Anforderungen beschrieben, welche das LCMS für den Einsatz im erlösorientierten Bildungsexport erfüllen sollte. Nach diesen Anforderungen wurden die in Absatz 2.3 betrachteten LCMS im folgendem Kapitel dahingehend analysiert, welche der Anforderungen bereits erfüllt werden, und welche Anforderungen als Erweiterung realisiert werden müssen.

3.1.1 Verwaltung von Lehrmitteln - Content Management

Alle betrachteten LCMS bieten Funktionen zum Ablegen und Weitergeben von Dateien und zum Zusammenstellen von Inhalten nach verschiedenen Gesichtspunkten. Teilweise ist es auch bereits möglich, Metadaten zu erfassen, nach diesen zu recherchieren und über diese Funktionen Materialien zu finden und zusammenzustellen. Im Rahmen der Arbeit werden einige erweiterte Anforderungen für das Management von Inhaltsobjekten betrachtet.

3.1.1.1 Repository

Das Repository als Datenbank bietet also das Backend zur Speicherung, Verwaltung, Recherche und Verwendung von Lehrinhalten. Dort sind Metadaten zu allen vorhandenen Objekten gespeichert, welche über verschiedene Schnittstellen eingepflegt und abgerufen werden können. Ebenfalls können dort Inhalte direkt gespeichert sein, aber auch Verweise auf andere, externe Datenquellen. Der Zugriff auf das Repository erfolgt über verschiedene, erweiterbare Schnittstellen.

3.1.1.2 Zielgruppenorientierte Zusammenstellung von Inhalten

Lehrinhalte, welche im Repository abgelegt werden, sollten als möglichst kleine, wiederverwendbare Einheiten dort abgelegt werden. Dementsprechend muss es dann auch Funktionen geben, die Lehreinheiten entsprechend den Anforderungen von verschiedenen Zielgruppen (z.B. verschiedene Studiengänge, Fernstudium, etc.) zusammenzustellen. Dabei sollte es auch möglich sein, Abhängigkeiten zwischen einzelnen Modulen zu definieren (z.B. Modul X darf

erst begonnen werden, wenn Modul Y bearbeitet wurde, oder aber auch in Abhängigkeit der Ergebnisse eines Tests).

Solche Zusammenstellungen von Lernmodulen sollten sich dann ebenfalls wieder im Repository ablegen lassen, um diese beispielsweise als Vorlage für neue Kurse benutzen zu können.

Als weitere Anforderung bei der Zusammenstellung von Lehrinhalten muss die Frage berücksichtigt werden, was bei Änderungen der Basis-Lehrinhalte passiert. Dabei sollte es bei einer Änderung der Basis-Lerneinheit mehrere Möglichkeiten geben:

- Die Änderung wird sofort übernommen
- Die alten Inhalte bleiben in der Zusammenstellung erhalten, die Änderung kann aber manuell übernommen werden
- Eine Änderung der Originalinhalte wirkt sich nicht auf die Zusammenstellung auf

Weiterhin sollte es hier möglich sein, Lehrinhalte selektiv für einzelne Zielgruppen freizuschalten. Dies sollte beispielsweise zeitgesteuert möglich sein, oder aber bei der Erfüllung von festgelegten Voraussetzungen durch den Lernenden, beispielsweise der Lösung von Aufgaben oder dem Bestehen von Tests.

3.1.1.3 Verwaltung von Metadaten

Zu jeder atomaren Lerneinheit, aber auch jeder Lektion oder jedem Kurs sollten sich Metadaten erfassen lassen, die zum Beispiel den Titel, eine Beschreibung, den Namen des Autors, Lernziele, inhaltliche und technische Voraussetzungen und weitere Daten enthalten sollten.

Diese Metadaten sind Voraussetzung für den einfachen Austausch der Lerneinheiten, Aufnahme in Kataloge und ermöglichen das systematische Durchsuchen der Daten.

Die Metadaten sollten dabei in einer standardisierten Form vorliegen oder aber sich in eine solche überführen lassen.

3.1.2 Informationspräsentation

Die Informationspräsentation innerhalb eines LCMS dient dem Aufbereiten der gespeicherten Lehrinhalte für die Benutzung durch den Anwender. Dabei soll das LCMS in der Lage sein, verschiedene Ausgabeformen und Medien zu unterstützen. Neben der interaktiven Bearbeitung der Lehrinhalte im Webbrowser (und ggfs. optimiert auf einem mobilem Gerät) sind schon für eine einfache Druckausgabe zahlreiche Anpassungen nötig. Beispielsweise müssen interaktive Elemente, welche im Druck nicht dargestellt / verwendet werden können, mit entsprechenden erklärenden Texten ausgestattet werden (welche in den Metadaten erfasst sein sollten). Dies ist beispielsweise auch notwendig, um die Inhalte barrierefrei zu gestalten.

3.1.2.1 Transformation und Anzeige verschiedener Dokumenttypen

Das Lernsystem sollte in der Lage sein, verschiedene Arten von Dokumenten zu verwalten und darzustellen. Alle betrachteten Lernsysteme beherrschen die Darstellung der meisten Dateitypen im Webbrowser, bei den meisten Systemen ist zudem eine aufbereitete Druckansicht möglich.

Bei XML-basierten System sollte es prinzipiell möglich sein, die Inhalte in andere Formate zu überführen, um so beispielsweise eine an mobile Endgeräte angepasste Version ausgeben zu können, oder aber auch direkt Formate für den Datenaustausch zu erzeugen.

Ja nach Ausgabemedium sollte das Lernsystem daher in der Lage sein, die Inhalte für das Ausgabemedium passend zu transformieren. Neben dem Webbrowser als Ausgabemedium sollte es auch möglich sein, sich die Inhalte in ansprechender Form ausdrucken zu können oder sie auf einem PDA oder anderem mobilen Gerät betrachten zu können. Hier sollte es eine variable Schnittstelle geben, welche die Umwandlung in das jeweilige Ausgabeformat übernimmt.

3.1.3 Autorenwerkzeuge zur Erzeugung von Lehrinhalten

Ein Lernmanagementsystem sollte die Möglichkeit bieten, die einzelnen Lernmaterialien im System zu bearbeiten. Autoren sollten die Möglichkeit haben, neue Objekte zu erstellen,

Objekte zu bearbeiten und zu aktualisieren.

Idealerweise sollte dies sowohl direkt online auf der Plattform möglich sein, um kleine Anpassungen und Änderungen direkt vornehmen zu können, als auch mit Hilfe von externen Werkzeugen, um eine Konvertierung von existierenden Materialien zu erleichtern und die Erstellung komplexerer Lernmaterialien zu vereinfachen.

3.1.4 Zusammenstellung von Lernobjekten

Lernmaterialien werden oft in verschiedenen Kursen und Zusammenhängen verwendet. Dies trifft insbesondere auf Material zu, welches Grundlagen zu einem bestimmten Fachgebiet vermittelt. Ein LCMS sollte es ermöglichen, nach bereits vorhandenen Lernmaterialien zu recherchieren, beispielsweise über eine Stichwort- oder Volltextsuche oder anhand der erfassten Metadaten, um bereits erfasste Materialien zum gesuchten Thema innerhalb des Systems zugänglich zu machen. Es sollte möglich sein, vorhandene Lernobjekte in verschiedenen Zusammenhängen einbinden zu können.

Wird ein Lernobjekt in verschiedenen Kontexten verwendet, sollte es möglich sein, anzugeben, was bei Aktualisierung des Originalobjektes passieren soll. Meistens, aber nicht immer, ist ein automatisches Update bei einer Aktualisierung des Objektes wünschenswert. Sind Änderungen an der Kopie des Objektes nötig, sind hier ggfs. auch komplexere Aktualisierungsszenarien nötig, wie sie beispielsweise in Versionsverwaltungssystem wie Subversion ([PCSF04]) Verwendung finden.

3.1.5 Erzeugung einer Navigationsstruktur

Die im vorhergehenden Abschnitt beschriebene Dokumentverwaltung enthält noch keine Informationen über die logische Strukturierung der verschiedenen Lernobjekte.

Während innerhalb der einzelnen Lerneinheiten eine entsprechende Strukturierung und Navigation schon vorhanden sein sollte, existieren noch keine Verknüpfungen zwischen den einzelnen Lernobjekten.

Werden diese zu neuen Kursen zusammengestellt, ist es nötig, dass diese verschiedenen Be-

reichen, Lernzielen oder zeitlich zugeordnet werden können und sich Erläuterungen zu den einzelnen Materialien erfassen lassen sollten, um beispielsweise für diesen Kurs besonders wichtige Teile des Lernmaterials anzugeben.

Es sollte auch möglich sein, dynamische Navigationen zu erzeugen, welche es ermöglichen, erst bei Erreichen einzelner Lernziele weitere, darauf aufbauende Lernobjekte anzuzeigen.

3.1.6 integrierte Kommunikationsmöglichkeiten

Ein solches Teleteachingsystem sollte auch eine Komponente enthalten, welche die Kommunikation der einzelnen Teilnehmer untereinander und mit dem jeweiligen fachlichen Betreuer ermöglicht. Insbesondere sollten folgende Funktionen enthalten sein:

- moderierte Foren zu jeweils einer Lerneinheit
- Einrichtung von virtuellen Lerngruppen
- direkte Fragen an einen fachlichen Betreuer
- persönliche und ggfs. öffentliche Kommentare zu einzelnen Seiten
- Bearbeitung von Aufgabenstellungen, Abgabe von Lösungen

Daneben sollen auch vom Studenten beantwortete Tests vom Betreuer halbautomatisch ausgewertet werden können, und die korrigierte Version dem Studenten zur Verfügung gestellt werden.

3.1.7 Interaktion mit anderen Anwendungen

Ein LCMS sollte es ermöglichen, Inhalte und Verwaltungsdaten auch mit anderen, externen Anwendungen austauschen zu können, um eine Einbindung des LCMS in eine IT Infrastruktur zu ermöglichen. Neben Single Sign On, welches im Abschnitt 3.2.1 betrachtet wird, sollten auch Schnittstellen vorhanden sein, um beispielsweise bereits vorhandene Kursstrukturen direkt auf das LCMS abbilden zu können, Inhalte zu im- oder zu exportieren, Kurse zu verwalten oder automatisiert weitere Verwaltungsfunktionen ausführen zu können.

3.1.7.1 API zum externen Zugriff auf LCMS-Funktionen

Eine API (Application Programming Interface) stellt eine definierte Schnittstelle bereit, um Funktionen des LCMS aus anderen Programmen heraus anzusprechen. Der Vorteil von definierten APIs ist, dass diese sich im Allgemeinen auch bei Aktualisierungen des LCMS nicht ändern und gut dokumentiert sind. Daher muss bei Nutzung dieser API eine externe Anwendung nicht für jede Version des LCMS angepasst werden.

Die API eines LCMS sollte alle im vorhergehenden Abschnitt angesprochenen Funktionen umfassen, um nicht direkt in das LCMS eingreifen zu müssen. Dies würde es erlauben, externe, zusätzliche Module, welche beispielsweise das LCMS in vorhandene Verwaltungsstrukturen einbinden, direkt über die API zu realisieren.

Wünschenswert wäre eine API, welche nicht nur den Zugriff durch lokal auf dem Server installierte Anwendungen ermöglicht, sondern auch für autorisierte Clients über eine Remote-Schnittstelle, wie zum Beispiel SOAP.

3.1.7.2 API zur Erweiterung des LCMS

Viele LCMS lassen sich durch Erweiterungsmodule um neue Funktionen erweitern. Diese Module benutzen die internen API-Funktionen des LCMS, um auf die Daten des LCMS zuzugreifen und um häufig genutzte Funktionen zu realisieren. Diese Schnittstelle ist im Vergleich zur API für externe Zugriffe fester an die Version des LCMS gebunden, so dass LCMS Erweiterungsmodule bei Erscheinen einer neuen LCMS-Version meist an diese angepasst werden müssen. Dafür ist diese Schnittstelle meist umfangreicher als die API für externe Zugriffe und ermöglicht den direkten Zugang zu den internen Daten des LCMS, und eine direkte Integration von neuen Funktionen ins LCMS, ohne das LCMS selbst zu ändern.

Das LCMS sollte eine solche, ausreichend dokumentierte Schnittstelle anbieten, um fehlende Funktionalität nachrüsten zu können.

3.1.7.3 Synchronisation von Inhalten bei Änderungen

Um eine Synchronisation der Inhalte zu betrachten, muss vorab analysiert werden, welche Ressourcen für eine Synchronisation zur Verfügung stehen und wie Teile oder komplette Einheiten von Inhalten übertragen werden können. Insbesondere sind hier Rahmenbedingungen wie beispielsweise die zur Verfügung stehende Bandbreite, Häufigkeit von Änderungen bei einzelnen Inhalten und die interne Struktur der Inhalte.

Zur Spezifikation der Schnittstellen wurden dabei, wie im Kapitel 4.6 beschrieben, verschiedene Varianten betrachtet, die den Datenaustausch zwischen verschiedenen Systemen ermöglichen.

Auf Basis eines solchen standardisierten Protokolls ist eine Form für die zu übertragenden Inhalte zu definieren, welche eine (ggfs. auch differentielle) Synchronisation von Inhalten zwischen verschiedenen Plattformen ermöglicht.

3.2 Sicherheitsanforderungen

In diesem Abschnitt sollen die Anforderungen an das LCMS hinsichtlich der Sicherheit und des Datenschutzes betrachtet werden.

3.2.1 Authentifizierung

Die Authentifizierung dient dazu, festzustellen, *wer* ein System benutzt. Dafür können u.A. folgende Verfahren eingesetzt werden:

- Authentifizierung mit Benutzername und Passwort
- Authentifizierung mittels Benutzerzertifikaten
- Authentifizierung mit Chipkarte und PIN
- Authentifizierung mittels biometrischer Verfahren
- Authentifizierung anhand der eingesetzten Hardware

- Token-basierte Authentifizierung

Im einzusetzenden System ist es wichtig, dass entweder als Quelle der Authentifizierungsdaten auch externe Quellen möglich sind, oder aber eine Synchronisation der Authentifizierungsdaten möglich ist, um einen Abgleich / Austausch von Benutzerinformationen zwischen verschiedenen Systemen zu ermöglichen. Nur so lässt sich eine gemeinsame Anmeldung für verschiedene Systeme ("Single Sign On") realisieren.

3.2.2 Autorisierung

Die Autorisierung dient dazu, festzulegen, auf welche Ressourcen der aktuelle Benutzer zugreifen darf und welche Operationen (Lesen, Ändern, etc) er mit diesen Ressourcen durchführen darf. Dabei können zusätzlich zu den Daten des Benutzers auch weitere Daten, beispielsweise der Standort des Rechners, in die Zuordnung der Rechte einfließen.

Die Zuordnung der Rechte zu einzelnen Ressourcen sollte einerseits möglichst detailliert für verschiedene Benutzergruppen möglich sein, andererseits aber durch die Verwendung von Vorlagen, Gruppen und Templates einfach zu konfigurieren und anzuwenden sein.

3.2.3 Schutz persönlicher Daten

Bei Betrieb eines E-Learning Systems fallen immer auch benutzerbezogene persönliche Daten an, da es für die meisten Systeme zwingend erforderlich ist, dass ein Benutzer einen persönlichen Account benutzt, um auf die Anwendung zugreifen zu können. Bei vielen Systemen ist dazu die Angabe einer E-Mail Adresse zwingend technisch erforderlich, beispielsweise um Benachrichtigungen oder Ankündigungen versenden zu können. Weitere persönliche Daten des Lernenden werden in vielen Systemen ebenfalls (gegebenenfalls optional) gespeichert. Zusätzlich werden bei LCMS - Systemen im laufenden Betrieb Status- und Bewertungsdaten erzeugt, beispielsweise welche Kapitel ein Student bereits bearbeitet hat, welche Tests er mit welchen Ergebnissen absolviert hat.

Diese Daten müssen im Sinne des Bundesdatenschutzgesetzes als personenbezogene Daten behandelt werden, daher ist eine Verarbeitung der Daten nur mit Zustimmung des Betroffenen

möglich. Ein LCMS sollte den Benutzer genau mitteilen, welche Daten für welche Personengruppen (Administratoren / Lehrende / Kursteilnehmer / Alle) einsehbar sind und auch die anonymisierte Erstellung / Darstellung der Daten ermöglichen, insbesondere bei der Darstellung von Testergebnissen und Umfragen.

3.2.4 Sicherstellung des Schutzes der Inhalte

Im Kapitel 2.8.4.2 wurden bereits etablierte Verfahren zum Schutz digitaler Inhalte vorgestellt. Im Weiteren sollen einige weitere Verfahren, welche im Rahmen der eingesetzten Java-Applets zur Anwendung kommen können, betrachtet werden.

3.2.4.1 Wasserzeichen

Vor der Auslieferung der Inhalte an den Benutzer werden diese personalisiert. Beispielsweise wird eine zusätzliche Datei mit in das Archiv aufgenommen, in einer vom Archiv bearbeiteten Datendatei ein entsprechender Vermerk ergänzt oder aber auch direkt in einer Klasse ein entsprechender Vermerk aufgenommen. Dieser Vermerk muss natürlich nicht im Klartext gespeichert werden, sondern kann auch verschlüsselt abgelegt werden, um eine Entdeckung und Entfernung des Vermerks zu verhindern.

Dieser Vermerk enthält Informationen, welche einen bestimmten Benutzer eindeutig identifizieren, beispielsweise die Benutzer-ID des LCMS Systems. Im Falle einer unberechtigten Weitergabe ist es dann zumindest möglich, festzustellen, welcher Benutzer die Datei weitergegeben hat.

3.2.4.2 Ausführungsbeschränkungen

Werden aktive Inhalte, wie Java-Applets oder Programme als Lehrmaterialien verwendet, so ist es möglich, innerhalb dieses Programms eine Prüfung der Authentifizierung und Autorisierung durchzuführen. Um doppelte Eingaben von Benutzerdaten zu vermeiden, kann hier auch eine Token-basierte Authentifizierung über das jeweilige LCMS erfolgen.

3.2.4.3 DRM Systeme für Medien

Zum Schutz von Video- und Audiodateien können, wie im Abschnitt 2.8.4.2 beschrieben, DRM Systeme eingesetzt werden, welche die Inhalte verschlüsselt ausliefern und erst mit einer entsprechenden Lizenz im jeweiligen Wiedergabeprogramm entschlüsseln. Dabei sollte gewährleistet sein, dass diese Systeme im Hintergrund arbeiten und entsprechende Lizenzen über eine Ankopplung an das LCMS System automatisch vergeben werden.

3.2.5 Übertragungssicherheit

Die Übertragung der Daten zwischen Server und Client, zumindest aber die Übertragung sensibler Daten wie Anmeldeinformationen und geschützte Inhalte, muss gegen ein Abhören gesichert sein. Zusätzlich ist es möglich, hier schon sicherzustellen, dass die Authentifizierung gegen den richtigen Kommunikationspartner erfolgt, um "Man-in-the-middle" Angriffe zu verhindern.

3.2.6 Weitere Anforderungen

- Leichte Anpassung an eigene Erfordernisse
- Erstellung von Lehrmaterialien direkt online ohne den Einsatz externer Programme möglich
- detaillierte Einstellung von Benutzerrechten, um es beispielsweise auch Studenten zu ermöglichen, in ausgewählten Bereichen Lehrmaterialien zu erstellen
- Verwaltung von Kursteilnehmern, Tutoren und Lerngruppen
- Verwaltung von Lehrmitteln
- Verwaltung von Terminen für die zeitgesteuerte Ausgabe von Lehrmaterialien, Aufgabenstellungen, Einreichung von Lösungen und Praktikums- oder Präsenzveranstaltungen

Diese Anforderungen werden von den im Abschnitt 2.1 betrachteten Systemen, teilweise mit einigen Einschränkungen, erfüllt. In dieser Arbeit sollen weitergehende Möglichkeiten von LCMS betrachtet werden, welche in den folgenden Abschnitten aufgeführt werden.

3.3 Technische Anforderungen an Lehrinhalte

An die Lehrinhalte sollten folgende Anforderungen gestellt werden:

- atomare Lerneinheiten, welche eine einfache Rekombinierbarkeit und Wiederverwendbarkeit ermöglichen
- Auszeichnung der Inhalte durch Metadaten
- semantische Auszeichnung der Inhalte
- zusätzliche Erschließung von Multimediainhalten/Simulationen durch alternative Texte

3.4 Datenaustausch mit anderen Systemen

In diesem Abschnitt werden verschiedene Möglichkeiten dargestellt, um Daten mit anderen Systemen, beispielsweise zwischen verschiedenen LCMS Systemen oder LCMS und externen Anwendungen, automatisiert auszutauschen. Dabei kommen die im Abschnitt 3.1.7 dargestellten technischen Schnittstellen zum Einsatz.

3.4.1 eSyndication

Auf Basis der in Abschnitt 2.4 dargestellten Protokolle soll ein Austausch der Metadaten und ggfs. auch einzelner Lerninhalte zwischen verschiedenen LCMS Systemen und ggfs. auch anderen Clients, welche das entsprechende Protokoll unterstützen, ermöglicht werden. Dabei wurden folgende Realisierungen betrachtet:

- Übertragung von aktuellen Informationen

- Vermitteln von Suchanfragen über Systemgrenzen hinaus
- Synchronisation von Lerninhalten (Repository-Objekten)
- Synchronisation von weiteren Inhalten (Forenbeiträge, etc.)

Die Kommunikation soll dabei auch bidirektional zwischen mehreren LCMS-Systemen möglich sein.

3.4.2 Automatischer Eintrag in Verzeichnisse- / Kataloge

Auf Basis der Metadaten und einer entsprechenden Kategorisierung soll eine Darstellung verfügbarer Lerninhalte auf zentralen Katalogplattformen, wie beispielsweise dem Bildungsportal Thüringen ([BPT], [FWH07], [WSK⁺02]), welche Inhalte aus vielen verschiedenen Quellen an zentraler Stelle sammeln, aufbereiten und nach verschiedenen Kriterien durchsuchbar anbieten, ermöglicht werden. Hier müssen allerdings evtl. gegebene Einschränkungen der Katalogplattform bezüglich der Metadaten betrachtet werden. Auch hier soll ein automatischer Austausch der Daten ermöglicht werden.

3.4.3 Verteilter Zugriff

Um eine Lastverteilung oder die Präsentation an verschiedenen Standorten mit geringstmöglicher Netzbelastung zu ermöglichen, sollte es möglich sein, verschiedene Server für die Lieferung der Inhalte (insbesondere der speicherintensiven Inhalte, wie Applets oder Multimedia) zu definieren und automatisch nach dem Standort des Clients den günstigsten Server auszuwählen.

Um einen teilweise autonomen Betrieb an verschiedenen Standorten zu ermöglichen, sollte es möglich sein, mehrere unabhängige Systeme Offline/Online zu synchronisieren, z.B. um aktualisierte Inhalte an verschiedenen Standorten zur Verfügung zu stellen.

Kapitel 4

Möglichkeiten zur Realisierung der Anforderungen

In diesem Abschnitt werden Möglichkeiten zur Realisierung der Anforderungen gem. Kapitel 3 betrachtet sowie die verschiedenen Varianten hinsichtlich ihrer Eigenschaften genauer untersucht. Dazu werden im Abschnitt 4.1 ausgewählte LCMS in Bezug auf die aufgestellten Anforderungen untersucht und verglichen, um eines der Systeme als Ausgangsbasis für das zu realisierende Gesamtsystem auszuwählen. Auf Basis dieses Systems werden dann im folgenden Erweiterungen zur Erfüllung der Anforderungen, welche nicht direkt vom LCMS bereitgestellt werden, untersucht.

Eine Übersicht der einzelnen Komponenten ist in Abbildung 4.1 dargestellt. Im Bild sind die neu realisierten bzw. erweiterten Komponenten rot dargestellt. Zusätzlich wurde eine Auswahl der untersuchten LCMS Komponenten dargestellt, diese ist jedoch, aufgrund der hohen Anzahl der Komponenten, nicht vollständig.

4.1 LCMS

Zur Grobauswahl des LCMS Systems wurden verschiedene Vergleiche in der Literatur verwendet ([JB], [BHH03], [Sch]), welche die LCMS Plattformen nach verschiedenen Kriterien wie didaktischen, organisatorischen und administrativen sowie ökonomischen Aspekten bewerten und vergleichen. Die Auswahl der LCMS wurde dabei, um die weiteren Anforderungen realisieren zu können, auf Systeme eingeschränkt, deren Quelltext unter einer Open Source Lizenz frei verfügbar ist und die unter einer Linux/PHP/MySQL Systemumgebung verwendet werden können. Diese Systeme wurden hinsichtlich der in Kapitel 3 aufgestellten technischen

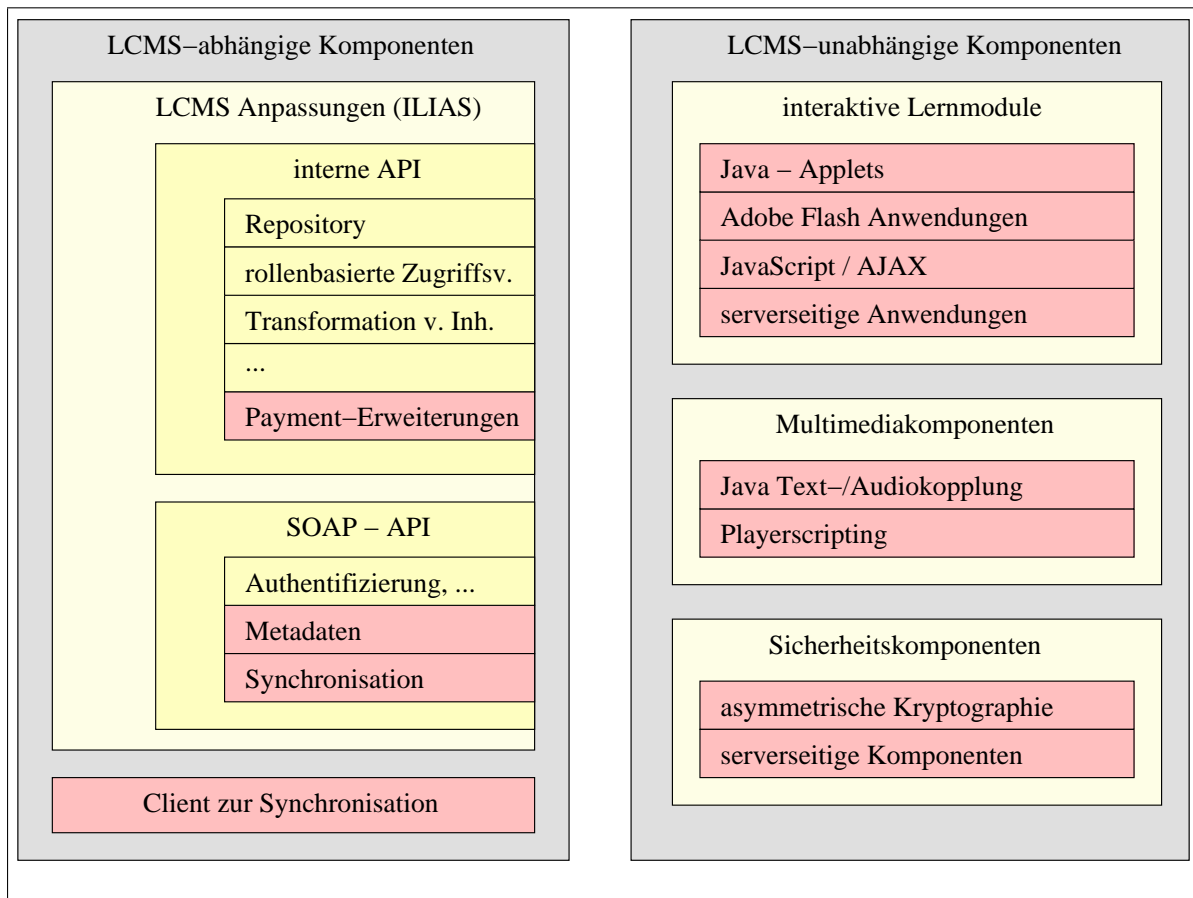


Abbildung 4.1: Übersicht über die realisierten Komponenten

Anforderungen untersucht. Dabei wurde neben der Benutzeroberfläche und der Funktionalität, welche in der Literatur die Basis der Vergleiche darstellen, auch der Quelltext und die Dokumentation der API mit in den Vergleich einbezogen, um eine Bewertung der internen API und die Realisierbarkeit von Erweiterungen zu ermöglichen.

Eine allgemeine Darstellung dieser LCMS wurde schon in Kapitel 2.3 vorgenommen, so dass in diesem Vergleich lediglich die Eigenschaften dargestellt werden, welche sich aus den Anforderungen nach Kapitel 3 ergeben. Die Tabelle 4.1 dient dabei der Übersicht, im den folgenden Abschnitten werden die einzelnen Punkte detailliert erläutert.

Tabelle 4.1: Vergleich der LCMIS Systeme bzgl. der Anforderungen

System	metacoorn	ILIAS	moodle
Version	3.8	3.9	1.8
Verwaltung von Lehrmitteln			
Repository	ext. Programm, zentrale Dateiablage	ja	nein
(Neu-) Zusammenstellung von Inhalten	ja	ja	innerhalb eines Kurses
Verwaltung von Metadaten	nur mit Zusatzprogramm	ja	nein
Transformation der Inhalte	PC, PDA (teilweise), XML extern	PC, Druck, XML online und extern	PC
Autorenwerkzeuge	extern	online und extern	online
interne Kommunikation	ja	ja	ja
Datenaustausch mit anderen Systemen			
API-Zugriff	-	SOAP	XML-RPC
eSyndication	nein	teilweise (News, Foren)	teilweise (News, Foren)
Synchronisation	-	-	-
verteilter Zugriff	-	-	-
interaktive, multimediale Anwendungen			
Unterstützung für Video & Audio	ja	ja	über Plugin
Synchronisation von Video mit anderen Inhalten	ja	nur indirekt	nur indirekt
Sicherheitsanforderungen			
Authentifikation	intern und extern (LDAP, RADIUS, weitere)		
Rollen-/Gruppenbasierte Autorisierung	ja (Unix-ähnlich)	ja, detaillierte Rechte	ja, detaillierte Rechte
Zugriffsbeschränkungen (IP-basiert)	nein	ja (Benutzer & Tests)	nein
Wasserzeichen/DRM	nein	nein	nein
anonymer Zugriff auf Inhalte	nein	ja	ja
Lernfortschritt anonym / personalisiert	n.V.	ja / ja	ja / ja
Logfiles anonymisierbar	nein	ja	nein
pseudonyme Teilname an Umfragen	nein	ja	nein

4.1.1 Repository, Verwaltung von Metadaten

Metacoon bietet zur Verwaltung und zum Zugriff auf das Repository ein externes Programm an. Nur mit diesem Programm ist es auch möglich, Metadaten zu den einzelnen Inhalten zu erfassen. Dies jedoch auf sehr flexible Art und Weise, das Programm unterstützt beliebige, konfigurierbare Sätze von Metadaten, Standards wie LOM oder Dublin Core werden unterstützt. Direkt auf der Webplattform ist ein Zugriff auf diese Daten oder ein Durchsuchen der Daten leider nicht vorgesehen, dort können nur die entsprechenden Inhalte zu Lerneinheiten arrangiert werden.

ILIAS verwaltet alle Inhalte in einer Baumstruktur, welche über das Webinterface zugänglich ist. Zu jedem Inhalt, angefangen von einzelnen Seiten über komplette Kapitel bis zu eingebundenen Medien lassen sich Metadaten nach dem LOM Standard hinterlegen. Diese sind auch direkt auf der Plattform abrufbar und durchsuchbar.

Moodle unterstützt das Ablegen von Metadaten nicht.

4.1.2 Zusammenstellung von Inhalten

Bei Metacoon ist es möglich, Inhalte aus dem Repository in einem Online-Editor zu Lerneinheiten zusammenzustellen. Der Online-Editor ermöglicht dabei das Zusammenstellen, Umpriorisieren und das hierarchische Anordnen der Inhalte. Da es bei Metacoon leider nicht möglich ist, Inhalte direkt auf der Plattform zu erzeugen, ist es leider auch nicht möglich, Zwischenüberschriften, Ergänzungen oder Hinweise direkt in die Lernmaterialien einzufügen. Dies kann nur über die integrierten Kommunikationsmodule, wie z.B. die Pinnwand erfolgen. Diese dient auch als zentrale Anlaufstelle im Kurs und von dort werden dann die jeweiligen Lernmaterialien verlinkt und gestartet. Die Einrichtung eines Kurses in metacoon ist aufgrund der komplexen Rechtevergabe und Auswahl der benötigten Module kompliziert. Bereits beim Erstellen des Kurses/Kursraumes müssen die benötigten Module festgelegt und eingebunden werden. Dies wird zwar mit Hilfe von Vorlagen erleichtert, sollen aber Kursräume angelegt werden, bei welchen keine der angebotenen Vorlagen passt, ist die Zusammenstellung der Module kompliziert.

Das Vorgehen zum Zusammenstellen von Lerneinheiten bei ILIAS unterscheidet sich je nach Art der Zusammenstellung. Das Zusammenstellen von Lerneinheiten, Aufgaben- und Kommunikationsmodulen sowie Tests zu einem Kurs ist sehr einfach. ILIAS unterstützt dabei auch das Anlegen von Verknüpfungen, was es bei Lehrmaterialien ermöglicht, die Lehrmaterialien, welche in mehreren Kursen verwendet werden, bei einer Aktualisierung der Inhalte automatisch in allen Kursen zu aktualisieren, wo diese verwendet werden.

4.1.3 Verwaltung von Metadaten

Metacoon bietet direkt auf der Plattform keine Eingabe und Verwaltung von Metadaten, welche über ein Beschreibungsfeld hinausgehen, an. Allerdings gibt es einen zusätzlichen Client auf Java-Basis, welcher zur Verwaltung des Repositories verwendet werden kann. Innerhalb dieses Clients ist es möglich, Metadaten nach beliebigen, konfigurierbaren Standards anzulegen, welche dann als XML-Datei im Repository abgelegt werden.

ILIAS unterstützt für alle Elemente im Repository Metadaten nach dem LOM Standard. Die Eingabe erfolgt direkt beim Anlegen oder Bearbeiten der Inhalte im Webinterface. Die wichtigsten Elemente werden dabei zusammengefasst auf einer extra Seite präsentiert, es können aber alle im Standard enthaltenen Daten angelegt werden. Es ist auch, wie bei LOM vorgesehen, möglich, Metadaten mehrsprachig zu erfassen.

Moodle unterstützt das Anlegen von Metadaten nicht.

4.1.4 eSyndication

Während Metacoon keine Möglichkeit bietet, Inhalte an andere Systeme weiterzugeben, ist es bei den anderen 2 betrachteten Systemen möglich, Inhalte im RSS-Format an andere Systeme zu übergeben. Dabei werden sowohl von ILIAS als auch von moodle die Weitergabe von Foreninhalten und News unterstützt.

Leider unterstützt keines der betrachteten Systeme die automatische Weitergabe von kompletten Lehrinhalten oder zumindest den Metadaten, hier ist jeweils nur eine Im-/Exportfunktion vorgesehen.

4.1.5 Transformation der Inhalte

Die Anpassung der Inhalte an verschiedene Plattformen ist nur in Metacoon und ILIAS realisiert. Metacoon bietet dabei eine auf die Darstellung auf kleinen Displays optimierte Oberfläche. Bei ILIAS ist es möglich, eine spezielle, konfigurierbare Druckversion von den einzelnen Inhalten zu erzeugen. Zum Im- und Export lassen sich die Daten bei ILIAS in verschiedene XML-Formate, unter anderem auch SCORM, transformieren. Moodle unterstützt an dieser Stelle nur ein eigenes XML-Format.

4.1.6 Autorenwerkzeuge

Metacoon unterstützt online nur das Anlegen von wenigen Inhalten, die eher in den Bereich “Kommunikation” einzuordnen sind, wie beispielsweise Pinwandbeiträge, und die Ablage von Dateien. Lerninhalte müssen vorab mit OpenOffice erstellt werden, mit einem speziellen Plugin in das Metacoon-Format konvertiert und auf die Plattform geladen werden. Dort können dann mehrere solcher Inhalte zu einem Kurs zusammengestellt und im entsprechenden Kursraum eingebunden werden.

ILIAS Lerneinheiten können direkt online erstellt werden, mit Hilfe eines komfortablen Online-Editors können neue Lerneinheiten, Aufgaben, Tests und weitere Inhalte direkt online erstellt werden. Zusätzlich ist es auch möglich, ILIAS-Lerneinheiten offline in OpenOffice zu erstellen und mit Hilfe eines Plugins in das ILIAS-Format zu konvertieren, diese sind nach dem Import auch online weiter bearbeitbar.

Moodle unterstützt nur die Erstellung von Inhalten online auf der Plattform, abgesehen vom Ablegen externer Dateien und dem Import von SCORM- und HTML-Lerneinheiten ist eine externe Erstellung von Inhalten nicht vorgesehen. In Moodle ist es mit Hilfe eines Rich Text Editors sehr einfach möglich, einfache Seiten, Aufgaben und weitere Inhalte zu erstellen. Bei der Erstellung von Lektionen unterstützt Moodle vielfältige Navigationsmöglichkeiten, wodurch die Erstellung von Lektionen komplex ist und eine vorherige Planung der Seitenstruktur erfordert, da diese sich nicht in einer Übersicht darstellen lässt.

4.1.7 Interne Kommunikation

Alle Systeme ermöglichen den plattforminternen Austausch von Nachrichten. Moodle und metacoon setzen dabei eine ähnliche Oberfläche wie Instant Messenger ein, während ILIAS ein internes Mailsystem verwendet.

4.1.8 API für externe Zugriffe

ILIAS bietet eine umfangreiche SOAP-Schnittstelle an, welche den Zugriff auf viele Funktionen der ILIAS-API ermöglicht und direkt innerhalb von ILIAS dokumentiert ist.

Moodle ermöglicht das Verwalten von Benutzern und Kursen über eine XML-RPC Schnittstelle.

4.1.9 API für Module und Erweiterungen

Moodle bietet für Lernaktivitäten, Lernmaterialien, Blockelemente und Textfilter eine einfache Schnittstelle für Erweiterungen. Neue Module können sehr einfach erstellt werden und werden automatisch in Moodle eingebunden, sofern die Dateien in den passenden Ordnern abgelegt werden. Der Zugriff auf die Funktionen von moodle erfolgt über die Einbindung und Verwendung mehrerer zentraler Bibliotheken von moodle. Diese sind zwar sehr gut dokumentiert, aufgrund der fehlenden Objektorientierung aber etwas unübersichtlich. Die Verwaltung von ggfs. erforderlicher Datenbanktabellen, Updates, administrative Einstellungen, etc. ist direkt die API integriert, so dass diese Funktionen direkt im Administrationsbereich von Moodle zur Verfügung stehen. In Moodle 1.9 ist außerdem ein Ereignissystem geplant, welches es dann ermöglicht, in Erweiterungen auf interne Ereignisse von Moodle zu reagieren und dort eigenen Code zur Ausführung zu bringen.

ILIAS verwendet eine objektorientierte, gut dokumentierte API, die das Schreiben von eigenen Erweiterungen sehr unterstützt. Die einzelnen Module arbeiten streng nach dem Model-View-Controller (MVC)-Prinzip. Allerdings ist das Einbinden von eigenen Erweiterungen kompliziert, da es in ILIAS keine direkte Schnittstelle dafür gibt, sind immer Änderungen im ILIAS-Code selbst erforderlich. Dies wiederum kann bei Updates der ILIAS-Plattform problematisch

sein, da diese dann entsprechend wieder eingearbeitet werden müssen.

Viele Komponenten von metacoon stammen aus anderen Open Source Projekten und wurden unter der Oberfläche von metacoon zusammengeführt. Daher sind die einzelnen Module relativ verschieden, nur für einige zentrale Bestandteile, wie beispielsweise die Benutzerverwaltung, gibt es eine zentrale API. Metacoon hat keine direkte Schnittstelle für Erweiterungen, auch hier sind nur Änderungen direkt im Source des LCMS möglich.

4.1.10 Synchronisation, verteilter Zugriff

Alle betrachteten Plattformen unterstützen die Synchronisation zwischen mehreren Plattformen nur über den Umweg des Ex- und Imports der Materialien. Auch ein verteilter Zugriff zur Lastverteilung ist nicht ohne größeren Aufwand realisierbar.

4.1.11 Multimediaunterstützung

Bei allen betrachteten Plattformen ist es möglich, Audio- und Videoinhalte einzubinden und wiederzugeben. Auch die Einbindung anderer aktiver Inhalte, wie Flash- und Java-Applets ist bei allen Plattformen möglich.

Metacoon unterstützt dabei als einziges der betrachteten Systeme eine direkte Unterstützung von Video und dazu synchronisierten weiteren Inhalten, deren Anzeige an das Abspielen des Videos gekoppelt ist. Dazu stellt Metacoon eine Oberfläche bereit, die es erlaubt, nach Upload eines Videos weitere Inhalte anzulegen, welche mit einem entsprechenden Zeitstempel versehen werden können und damit synchron zum Video wiedergegeben werden können.

4.1.12 Informationspräsentation auf verschiedenen Plattformen

Die hier betrachteten LCMS Systeme sind vorrangig alle darauf ausgerichtet, Inhalte zu verarbeiten und zu präsentieren, welche mittels eines Webbrowsers und ggfs. einigen Plugins angezeigt werden können.

Moderne PDA und Mobiltelefone verfügen über die Möglichkeit, XHTML-konforme Webseiten

anzuzeigen. Allerdings müssen bei Lernmaterial für diese Geräte die Inhalte entsprechend der Displaygröße angepasst werden, insbesondere müssen beispielsweise Bilder verkleinert werden. Mobile Geräte können heute meist PDF-Dateien anzeigen, so dass dieses Format ebenfalls verwendet werden kann. Eine Darstellung auf mobilen Systemen ist im Moment nur bei metacoon realisiert. Bei ILIAS ist ein derartiger Ansatz zur Zeit in der Entwicklung. ILIAS unterstützt darüber hinaus eine konfigurierbare Druckausgabe, deren linearisierte XHTML-Ausgabe der Lehrinhalte sich auch auf einem PDA betrachten lässt. Allerdings ist das nicht mit einer PDA-Version gleichzusetzen, da für diese Funktion zuerst die aufbereiteten Inhalte am PC vorbereitet und auf den PDA transferiert werden müssen.

Während sich einzelne Seiten bei allen Lernplattformen für die Offline-Nutzung drucken lassen, bietet nur ILIAS die Möglichkeit, eine spezielle, anpassbare, optimierte Druckversion von einzelnen Materialien zu erzeugen.

4.2 Sicherheit und rechtliche Aspekte

Im Abschnitt 2.8 und den jeweiligen Abschnitten zu den LCMS-Systemen wurde schon auf die grundlegenden Mechanismen zur Sicherung der Übertragung und des Zugriffs auf die Inhalte eingegangen. Im Folgenden sollen einige weitere Maßnahmen zur Verhinderung der Weitergabe von geschützten Lerninhalten beschrieben werden.

4.2.1 Ausführungsbeschränkungen

Java Applets dürfen mit dem Server, von welchem sie geladen wurden, kommunizieren. Auf Basis dieser Kommunikation ist es möglich, Applets so zu gestalten, dass diese nur von einem bestimmten Server geladen werden dürfen. Technisch gibt es dazu mehrere Möglichkeiten, von einer einfachen Abfrage der IP des Ursprungsservers bis hin zu einer sicheren Identifizierung des Servers auf Basis asymmetrischer Kryptographie, bei welcher das Applet eine Zufallszahl zum Server übermittelt, der Server diese seinem geheimen Schlüssel signiert und das Applet die Signatur mit Hilfe des öffentlichen Schlüssel des Servers prüft.

4.2.2 DRM in Java

In den folgenden Kapiteln werden verschiedene Möglichkeiten zur Sicherung von Java-basierten interaktiven Inhalten dargestellt, und die Grenzen dieser Systeme erläutert.

4.2.2.1 Java-DRM mittels asymmetrischer Kryptographie

In [Tse06] wurde in Java ein DRM System auf Basis von Java und PGP entwickelt. Bei diesem System wird der eigentliche Inhalt verschlüsselt zum Client übertragen. Die Entschlüsselung erfolgt mit Hilfe einer Lizenzdatei, welche individuell für jeden Nutzer erzeugt wird (siehe Bild 4.2). Diese ist wiederum mit einem persönlichen asymmetrischen Schlüssel des Benutzers verschlüsselt. Um die Weitergabe dieses Schlüssels zu verhindern, kann hier auch eine von gnuPG unterstützte Smartcard zum Einsatz kommen.

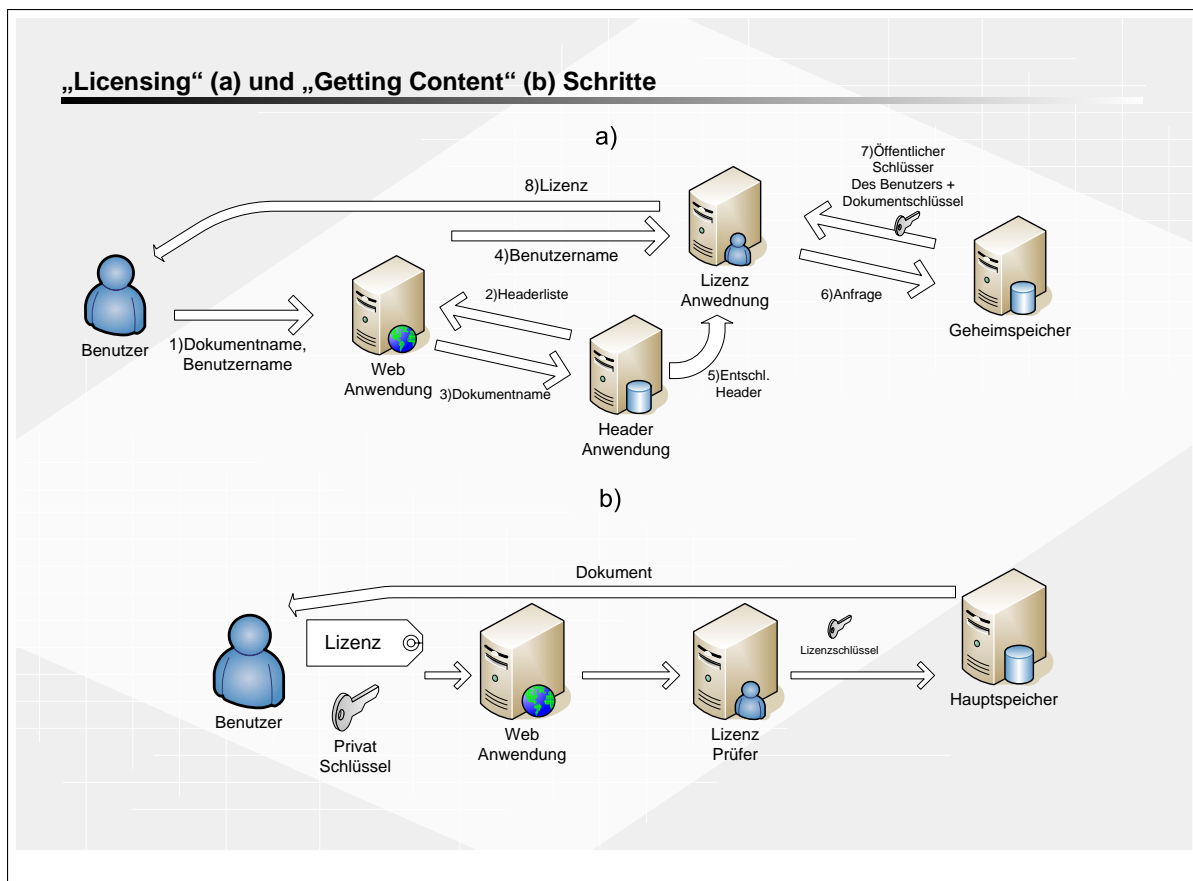


Abbildung 4.2: DRM in Java - Lizenzierung (aus [Tse06])

4.2.2.2 Auslagerung von Programmkomponenten auf Serverseite

Im Petri Netz Applet (siehe 5.5) wurde ein anderer Weg gewählt. Dort werden, einerseits wegen der einfacheren Realisierbarkeit, andererseits aus Gründen des Schutzes der Inhalte, einzelne Funktionen vom Server berechnet. Der Server kann dabei feststellen, welcher Client (IP-Adresse) die Daten anfordert und dementsprechend ggfs. bei einer unberechtigten Anfrage mit einer Fehlermeldung statt dem Ergebnis antworten. Im Fall des Petri Netz Applets wird dabei für die Funktion „Erreichbarkeitsgraph“ lokal vom Applet eine Datenstruktur berechnet, die die möglichen Zustände und Zustandsübergänge enthält. Diese Daten werden zum Server übertragen, welcher mit Hilfe externer Programme daraus eine grafische Darstellung erzeugt und als Ergebnis an das Java Applet zurückschickt.

4.2.2.3 Grenzen

In Java ist es durch die Verwendung eines plattformunabhängigen Bytecode relativ einfach möglich, Programme zu dekompileieren und damit wieder Quellcode zu erhalten. Dadurch ist es auch möglich, die oben angesprochenen Schutzmaßnahmen auf Clientseite zu umgehen. Zumindest als „rechtmäßiger“ Nutzer eines Programms hat man Zugriff auf alle auf dem Client ausgeführten Programmkomponenten und kann diese entsprechend auch manipulieren. Damit lässt sich beispielsweise der in Abschnitt 3.2.4.2 dargestellte Schutz einfach umgehen. Bei der Verwendung asymmetrischer kryptographischer Verfahren ist dies komplizierter, allerdings, beispielsweise durch den Austausch der verwendeten Schlüssel, auch nicht unmöglich. Nur bei serverunterstützten Verfahren sind die Teile, welche auf dem Server berechnet werden, relativ sicher, zumindest solange es keine Möglichkeit gibt, die Inhalte des Servers auszulesen, was aber im Normalfall gegeben ist.

4.3 Interaktive Inhalte und Anwendungen

Interaktive Inhalte und Anwendungen, wie Simulationen, Modelle und parametrisierbare Beispiele und Animationen bieten dem Lernenden die Möglichkeit, das theoretische Wissen anzuwenden. Dabei gibt es verschiedene Möglichkeiten, diese interaktiven Inhalte zu realisieren.

Im Abschnitt 2.6 wurden bereits die wichtigsten Technologien für eine Darstellung von interaktiven Inhalten direkt im Webbrowser dargestellt. Neben dieser Möglichkeit existieren noch weitere Möglichkeiten, Lehrmaterial mit interaktiven Elementen zu erstellen.

4.3.1 Interaktive Inhalte als Anwendung im Webbrowser

Um interaktive Inhalte direkt im Webbrowser darzustellen, sind die in Abschnitt 2.6 dargestellten Möglichkeiten weit verbreitet:

- Java Applets
- Adobe Flash
- JavaScript / AJAX
- serverseitige Anwendungen

Jede dieser Techniken hat Ihre Vor- und Nachteile, so dass es im Wesentlichen von den Anforderungen der Inhalte abhängt, welche Technik zum Einsatz kommt. Auch ein gemischter Einsatz verschiedener Techniken ist möglich.

Mit Hilfe von Java Applets ([Wut06]) lässt sich nahezu jede Funktion realisieren. Neben der Darstellung im Browser kann ein Java Applet auch mit dem Server, von welchem es geladen wurde, direkt kommunizieren. Wird das Java Applet signiert und der Anwender räumt dem Applet zusätzliche Rechte ein, kann ein Applet auch auf lokale Dateien zugreifen oder Daten mit fremden Servern austauschen. Für Java gibt es zahlreiche freie Bibliotheken, welche das Programmieren von Applets erleichtern und benötigte Funktionen bereits zur Verfügung stellen. Java Applets haben aber auch einige Nachteile: Aus Sicht des Erstellers ist es in Java vergleichsweise kompliziert, eigene grafische Oberflächen und Animationen zu erstellen, da alle Elemente auf Quelltextebene erstellt und verwaltet werden müssen. Die Standard-Grafikschnittstelle von Java ist nicht vektorbasiert, so dass man beispielsweise Anpassungen an die Ausgabegröße selbst vorsehen muss. Der Anwender muss meist erst die Java- Laufzeitumgebung installieren, welche relativ umfangreich ist.

Mit Hilfe von Adobe Flash lassen sich sehr einfach Animationen und andere grafische Elemente erstellen. Auch die Wiedergabe von Audio und Video wird direkt unterstützt. Die Darstellung ist dabei zu großen Teilen vektorbasiert und frei skalierbar. Die Entwicklungsumgebung von Adobe Flash ist dabei ähnlich einem Mehrspureditor für den Videoschnitt aufgebaut. Auf den einzelnen Spuren können innerhalb von Keyframes grafische Elemente, welche auch aus einer umfangreichen Bibliothek entnommen werden können, platziert werden. Jedes Objekt lässt sich zwischen den Keyframes auf einfache Art und Weise animieren. Alle Aktionen können auch programmgesteuert, ggfs. basierend auf den Aktionen des Nutzers, mit Hilfe der Programmiersprache ActionScript ausgelöst werden. Natürlich sind mit Hilfe von ActionScript auch das Verarbeiten von Eingaben, komplexe Berechnungen und die Kommunikation mit dem Server möglich, da ActionScript ab Version 2 eine vollständige, objektorientierte, Java-ähnliche Programmiersprache ist. Nachteile von Flash sind die enge Kopplung von Scripten an Frames, was längere Scripte schnell unübersichtlich werden lässt. Zudem ist die Entwicklungsumgebung für Flash nicht frei verfügbar.

Mit Hilfe von JavaScript bzw. AJAX ist es möglich, programmgesteuerte Aktionen innerhalb einer Webseite auszuführen. Dabei kann JavaScript beispielsweise dazu eingesetzt werden, um Eingaben zu überprüfen, Berechnungen nachzuvollziehen, etc., aber auch, um auf Aktionen des Lernenden zu reagieren und die angezeigte Seite dynamisch zu verändern. Mit Hilfe von AJAX ist es möglich, im Hintergrund Daten zu einer Serveranwendung zu übertragen und die vom Server gelieferten Ergebnisse entsprechend zu präsentieren. JavaScript ist leider nicht in der noch nicht Lage, selbst grafische Elemente programmgesteuert zu erzeugen, so dass sich grafische Anwendungen nur mit erheblichem Aufwand und serverseitiger Hilfe realisieren lassen.

Serverseitige Anwendungen erlauben es, basierend auf Eingaben des Benutzers verschiedene Abläufe und Aktionen durchzuführen. Während die Ein- und Ausgabe weiterhin über den Browser läuft, und man daher an dessen Möglichkeiten gebunden ist, können auf der Serverseite beliebige Programme eingesetzt werden, um die Eingaben des Nutzers zu verarbeiten und Rückmeldungen zu erzeugen. Auch ein Ansteuern von externer Hardware, welche mit dem Server verbunden ist, ist so möglich. Im Zusammenspiel mit den oben genannten clientseitigen Anwendungen ist es auch möglich, grafische Ausgaben oder komplexere Anwendungen

zu realisieren. Vorteile der Realisierung auf dem Server sind, dass man auf dem Server beliebige Programme benutzen kann, sofern sich die Ein- und Ausgaben so umformen lassen, dass sie zur Eingabe und Darstellung im Webbrowser geeignet sind, und dass die eigentliche Anwendung nie den Server verlässt. Nachteile sind die relativ komplizierte Einrichtung auf dem Server und, bei einer größeren Zahl gleichzeitiger Benutzer, die hohe Rechenlast für den Server.

Gemischte Anwendungen erlauben es, die Vorteile von clientseitiger Programmierung mit den Vorteilen von serverseitigen Programmen zu verbinden. Dabei wird ein Großteil der Datenumformung / Darstellung / etc. auf dem Client durchgeführt, und nur Schritte, welche sich beispielsweise anders nur mit externen Programmen realisieren ließen, werden dem Server übertragen.

4.3.2 Interaktive Inhalte mit Hilfe externer Programme

Eine weitere Möglichkeit für interaktive Lehrinhalte besteht darin, Aufgaben, interaktive Arbeitsblätter und Materialien im Format einer externen Anwendung zur Verfügung zu stellen. Je nach Lizenz kann zusätzlich diese externe Anwendung ebenfalls mit angeboten werden. Dies erlaubt umfangreichere und komplexere Lernmaterialien, da alle Funktionen der externen Anwendung, wie beispielsweise eine Programmierumgebung oder ein Computeralgebrasystem, benutzt werden können, welche für eine Simulation im Browser zu komplex wären.

Lässt die Lizenz des Programmes eine Weitergabe nicht zu oder wird spezielle Hardware dazu benötigt, besteht auch die Möglichkeit, das externe Programm auf dem Server ablaufen zu lassen und über das Internet fernzubedienen. Dazu gibt es mehrere Möglichkeiten, beispielsweise bei Microsoft das „Remote Desktop Protokoll“ (RDP), unter Linux X11, oder, plattformübergreifend, VNC (von welchem es zahlreiche Varianten gibt). Während bei RDP der entsprechende Client auf dem PC des Lernenden installiert sein muss, kann VNC auch über ein Java-Applet direkt im Browser benutzt werden. Da das Programm in diesem Fall auf dem Server läuft, ist es nötig, den Zugriff auf diese Schnittstelle einzuschränken und zu steuern, so dass nur eine festgelegte Anzahl Benutzer das System jeweils gleichzeitig benutzen können. Diese Aufgabe kann direkt vom LCMS durchgeführt werden. Bei Moodle gibt

es dazu beispielsweise das Plugin „Booking“. Allerdings wird zur Übertragung der Daten in diesem Fall eine schnelle Internetverbindung benötigt und nicht alle Programme lassen sich auf diese Art steuern, da es bei schnellen Bildwechseln (Animationen, Videos) durch die Netzwerkübertragung zu Verzögerungen kommt.

4.4 Formate und Formatkonvertierungen

Im Abschnitt 2.5 wurden bereits die wichtigsten standardisierten Formate für Lehrinhalte und Metadaten vorgestellt. In diesem Abschnitt sollen nur die Formate für Lehrinhalte betrachtet werden. Neben den standardisierten Formaten SCORM, AICC und IMS (QTI) werden natürlich auch weitere, universelle Formate für Lehrinhalte eingesetzt, welche hier mit betrachtet werden sollen.

4.4.1 HTML/XHTML und HTML-basierte Formate

HTML/XHTML und darauf aufbauende Formate wie AICC und SCORM finden eine weite Verbreitung für Lerninhalte. Durch die direkte Darstellung im Browser und den zahlreichen vorhandenen Werkzeugen ist es sehr einfach, entsprechende Inhalte zu erstellen. Es lassen sich weitere Komponenten direkt referenzieren, welche dann mit Hilfe des entsprechenden Plugins dem Lernenden direkt angezeigt werden.

Allerdings hat die direkte Erstellung von Lerninhalten in HTML auch einige Nachteile. So ist es nicht ohne weiteres möglich, geschlossene Lerneinheiten zu definieren, welche sich trotzdem zu komplexeren Kursen zusammensetzen lassen. In HTML eingebettete Metadaten (nach dem Dublin Core Standard) lassen sich nur relativ kompliziert automatisch erfassen, auch ist nicht festgelegt, auf welcher Ebene diese Daten erfasst werden. HTML/XHTML unterstützt als Textauszeichnungssprache keine Interaktivität und keine Kommunikation mit dem LCMS, diese lässt sich jedoch durch das Referenzieren aktiver Komponenten oder durch den Einsatz von JavaScript nachrüsten. Diese angesprochenen Probleme wurden durch den SCORM-Standard weitgehend gelöst, indem dort die zusätzlich nötigen Daten in XML-Dateien ergänzt werden und eine Schnittstelle zur Kommunikation mit dem LCMS definiert wird, wie in

Kapitel 2.5.4 beschreiben. Ein weiteres Problem bleibt allerdings: Es wird keine semantische Auszeichnung der Inhalte unterstützt. Damit ist es beispielsweise kompliziert, aus einer HTML-Lerneinheit alle Definitionen oder Beispiele zu extrahieren.

Die Konvertierbarkeit von HTML in andere Formate wird im Wesentlichen vom Aufbau der HTML-Daten, insbesondere der eingesetzten Formatierungsmittel und Auszeichnungen, selbst bestimmt. Enthält das HTML wie vorgesehen nur die reinen Inhalte, und die Formatierung wird über CSS vorgenommen, ist eine Konvertierung in andere Formate (beispielsweise mit htmldoc nach PDF), eine Anpassung an andere Bildschirmauflösungen, etc. relativ einfach. Wurde der HTML-Code aber beispielsweise von einem Anwendungsprogramm automatisch generiert, ist dieser sehr oft mit zusätzlichen reinen Layout-Elementen und Formatierungen versehen, welche eine Konvertierung sehr kompliziert werden lassen und zu unvorhergesehenen Ergebnissen führen können.

4.4.2 XML basierte Formate

XML basierte Formate bieten vor allem eine große Flexibilität, da sich XML relativ einfach, zum Beispiel mit Hilfe von XSLT, in andere Formate überführen lässt. Um Lerninhalte sinnvoll in XML darzustellen, benötigt man aber auch ein passendes XML Schema bzw. DTD, das die nötigen Elemente zur semantischen Auszeichnung enthält. Dies ist beispielsweise bei dem vom ILIAS verwendeten Format der Fall.

Zur Konvertierung von XML in andere Formate kommt meist XSLT zum Einsatz. Damit lässt sich XML auf relativ einfache Weise in andere Formate umwandeln, zum Beispiel in HTML/XHTML für die Darstellung im Browser. Es ist damit auch ein direkter Zugriff auf bestimmte Elemente möglich, was es bei geeignetem Schema ermöglicht, einzelne Elementgruppen zu extrahieren.

4.4.3 Postscript und PDF

Postscript wurde als Seitenbeschreibungssprache 1984 von der Firma Adobe entwickelt. Es ist eine vollständige, stack-orientierte Programmiersprache, die allerdings keine Interaktion mit

dem Anwender zulässt. Diese wurde mit PDF noch erweitert, so das Inhaltsverzeichnisse, Hyperlinks, Formulare und weitere Elemente möglich sind. Das Format ist für die Druckausgabe vorgesehen und durch Einbettung benötigter Schriften und weiterer Elemente dafür optimiert. Die Anzeige auf dem Bildschirm entspricht daher in den meisten Fällen dem Druckbild.

PDF Dokumente lassen sich nur unter großem Aufwand in andere Formate konvertieren. Es ist beispielsweise möglich, das Dokument oder einzelne Seiten als Bilder zu exportieren. Auch ist es in vielen Fällen möglich, den vorhandenen Text auf einem PDF-Dokument für eine Suchfunktion nutzbar zu machen. Dabei gehen aber Formate, Layout und semantische Informationen verloren.

Eine Konvertierung von vielen anderen Formaten in PDF ist dagegen relativ einfach, da dies beispielsweise über einen virtuellen Druckertreiber aus jeder Anwendung mit Druckfunktion möglich ist. Viele Programme, wie beispielsweise OpenOffice, unterstützen direkt die Ausgabe in PDF. Eine Konvertierung von HTML nach PDF ist unter anderem mit `htmldoc` möglich, für komplexere Dokumente besteht auch die Möglichkeit, diese ins LaTeX Format zu übertragen und mit `pdflatex` in ein PDF zu übersetzen.

4.5 Multimediaanbindung

Im folgenden Abschnitt werden Möglichkeiten behandelt, Text- und Bildinhalte und Simulationen um Audio- und Videomaterial zu ergänzen und eine Synchronisation der Medien mit der Anzeige von weiteren Inhalten, wie Texten, Grafiken und Animationen beim Abspielen zu ermöglichen.

4.5.1 Java Text/Audio Kopplung

Im Rahmen des Applets „interaktive Folien“ (siehe 5.3) wurde auf Basis der Klasse „BundleSupport“, welche für eine mehrsprachige Darstellung der Inhalte verantwortlich ist, eine Anbindung von gesprochenen Texten zur Erläuterung der dargestellten Prozesse erstellt.

Ausgangsbasis dafür waren bereits vorliegende Audioaufzeichnungen der Texte. Als Framework für die Wiedergabe wurde das Java Media Framework verwendet, welches eine platt-

formunabhängige Wiedergabe der Dateien ermöglicht. Ziel war es, die Wiedergabe der Audiodateien und das Abspielen von Animationen zu synchronisieren, so dass die jeweils zum Animationsschritt passende Erläuterung abgespielt wird.

Dazu wurde die Klasse „MediaManager“ entwickelt, welche, bei Vorliegen der nötigen Rahmenbedingungen sowie der Aktivierung der Audiowiedergabe im Applet, eingebunden wird. Diese Klasse wertet in einer sprachabhängigen Steuerdatei vorgegebene Einträge aus, die zu jedem Animationsschritt beinhalten, welche Audiodatei ab welcher Position abzuspielen ist. Über eine Benachrichtigungsfunktion wird nach der Wiedergabe die Animationsklasse „SimulationEngine“ vom Ende der Wiedergabe benachrichtigt, so dass diese ggfs. automatisch den nächsten Animationsschritt auslösen kann. Ebenfalls wurden in der entgegengesetzten Richtungen entsprechende Benachrichtigungsfunktionen eingebaut, so dass die Audiowiedergabe auch flexibel auf Klicks des Anwenders in der Simulation, beispielsweise das Stoppen der Animation oder die Auswahl eines Objektes, reagieren kann. Für längere Texte wurden Indexe implementiert, welche es ermöglichen, durch Anklicken von entsprechend (parallel dargestellten) Textteilen auch in der Audiowiedergabe an die entsprechende Stelle zu springen.

4.5.2 Video / Playerscripting

Es bietet sich an, Vorlesungen und andere Materialien als Videostreams zur Verfügung zu stellen. Da die Videostreams für die Übertragung über das Internet komprimiert werden müssen, reicht dabei meist die Qualität zur leserlichen Anzeige von Folien, Simulationen und anderen Inhalten nicht aus. Daher muss ein Weg gefunden werden, diese Medien auf anderer Art und Weise mit dem Videostream zu verknüpfen.

Im folgenden wird dies für Windows Media betrachtet, welches hier zur Erstellung der Inhalte eingesetzt wurde.

Windows Media erlaubt es, während der Codierung oder auch nachträglich Skriptbefehle in den Datenstrom einzubetten. Neben der Möglichkeit der Texteinblendung und dem Aufrufen von URLs in einem anderem Frame ist es auch möglich, dort beliebige benutzerdefinierte Kommandos einzufügen.

Wird der Windows Media Player innerhalb einer Webseite im Internet Explorer angezeigt,

werden verschiedene Event-Handler aufgerufen [WMPb]. Neben allgemeinen Handlern für Statusänderungen gibt es dort auch "ScriptCommand", welches beim Auftreten eines benutzerdefinierten Befehls im Datenstrom aufgerufen wird. Dabei wird ein (ebenfalls im Datenstrom eingebetteter) Parameter übergeben, welcher es ermöglicht, verschiedene Aktionen zu unterscheiden. So ist es durch entsprechende Programmierung des Event Handlers möglich, externe Foliengrafiken zu tauschen, entsprechende Aktionen in per Script steuerbaren Simulations-Applets auszulösen oder aber ggfs. auch den Player selbst zu steuern, um z.B. scriptgesteuert das Video zur Bearbeitung einer Aufgabe zu pausieren. Der eingebettete Windows Media Player selbst lässt sich dabei komplett per Skript steuern, neben den Standardfunktionen wie Wiedergabe / Pause / etc. ist beispielsweise auch das Ändern des aktuellen Streams oder das Anspringen von Marken möglich.

Um die Nachbearbeitung bei der Videoaufzeichnung einer Vorlesung zu vereinfachen, wurde ein Makro für Powerpoint erstellt, welches während der Aufzeichnung der Vorlesung im Hintergrund entsprechende Zeitindexe sammelt und in einer Datei speichert. Mit Hilfe dieser Datei (und der eigentlichen Folien, welche auch per Makro in der richtigen Auflösung und Dateiformat exportiert werden können) ist es sehr einfach, entsprechende Präsentationen, welche die einzelnen Folien synchronisiert zum Video anzeigen, zu erstellen.

Daneben existiert von Microsoft ein Programm Microsoft Producer([WMPa]) als Zusatz zu Microsoft Powerpoint, welches es ermöglicht, solche Präsentationen direkt zu erstellen. Der Producer erlaubt die Anordnung von Video, Powerpoint-Folien, Marken und Scriptbefehlen in einer Zeitleiste, wie das bei Videoschnittprogrammen üblich ist. Die nötigen Steuerdateien für eine Internet-Präsentation werden dann automatisch erzeugt.

4.6 Datenaustausch

Um einen Datenaustausch beim Einsatz von E-Learning im Bildungsexport zu ermöglichen, werden verschiedene Verfahren zur Synchronisation der Inhalte betrachtet. Da die verschiedenen Partner unterschiedliche Anbindung an das Internet haben, werden verschiedene Verfahren betrachtet, um den Datenaustausch zu realisieren. Dabei wird im folgenden davon ausgegangen, dass lokal ausreichend Bandbreite und Serverkapazität zur Verfügung steht, um

die Inhalte verarbeiten und bereitstellen zu können.

4.6.1 Bandbreitenmanagement

4.6.1.1 Anbindung des Partners mit hoher Bandbreite

Bei Anbindung der Partnerhochschule mit hoher Bandbreite ist es möglich, alle Inhalte nur lokal vorzuhalten und entsprechend zur Nutzung durch die Studenten auf Abruf zur Partnerhochschule zu übertragen. Vorteile dieses Verfahrens ist die einfache Verwaltung der Inhalte an einem Ort, schnelle Einarbeitung von Änderungen und eine zentrale Plattform zur Kommunikation.

4.6.1.2 Anbindung des Partners mit mittlerer Bandbreite

Ist eine Übertragung aller Inhalte auf Abruf aufgrund der zur Verfügung stehenden Bandbreite nicht möglich, sind Teile der Lernmaterialien auszulagern. Dabei kann das im Bild 4.3 dargestellte Verfahren zum Einsatz kommen: Während die Lernplattform selbst mit allen Kommunikations- und Planungsfunktionen lokal betrieben wird und auch Texte und andere Inhalte, welche nur eine geringe Bandbreite benötigen und sich häufig ändern nur lokal vorgehalten werden, werden Inhalte, welche eine höhere Bandbreite zur Übertragung benötigen, am Standort der Partnerhochschule vorgehalten und in der Lernplattform entsprechend verlinkt. Dabei können die Inhalte an der Partnerhochschule sowohl lokal auf den Clients vorgehalten werden, als auch auf einem dortigen Server abgelegt sein, so dass diese Inhalte nur im LAN übertragen werden müssen. Über eine entsprechende Konfiguration der Lernplattform werden alle Links für diese Inhalte an die Erfordernisse der Plattform der Partnerhochschule angepasst.

Inhalte, welche über dieses Verfahren an der Partnerhochschule vorgehalten werden, sollten sich nur selten ändern, wie beispielsweise Videomitschnitte von Vorlesungen und Seminaren, Animationen und Java Applets. Über eine der im Abschnitt 4.6.2 beschriebenen Technologien wird sichergestellt, dass die Inhalte, sollten diese geändert werden, aktualisiert werden.

Vorteile dieses Verfahrens sind die Nutzung einer zentralen Plattform zur Kommunikation

und Planung, und die einfache Verwaltung aller sich oft ändernden Inhalte bei gleichzeitiger ressourcenschonender Auslagerung von Medien auf den Client oder einen Server der Partnerhochschule. Nachteile dieser Methode sind die kompliziertere Administration der Inhalte, da je nach Partner und dessen Methode zur Speicherung der ausgelagerten Inhalte die entsprechenden Verweise innerhalb der Lehrmaterialien angepasst werden müssen, sowie der Aufwand der Einrichtung und Durchführung einer Synchronisation von ausgelagerten Inhalten bei Änderungen.

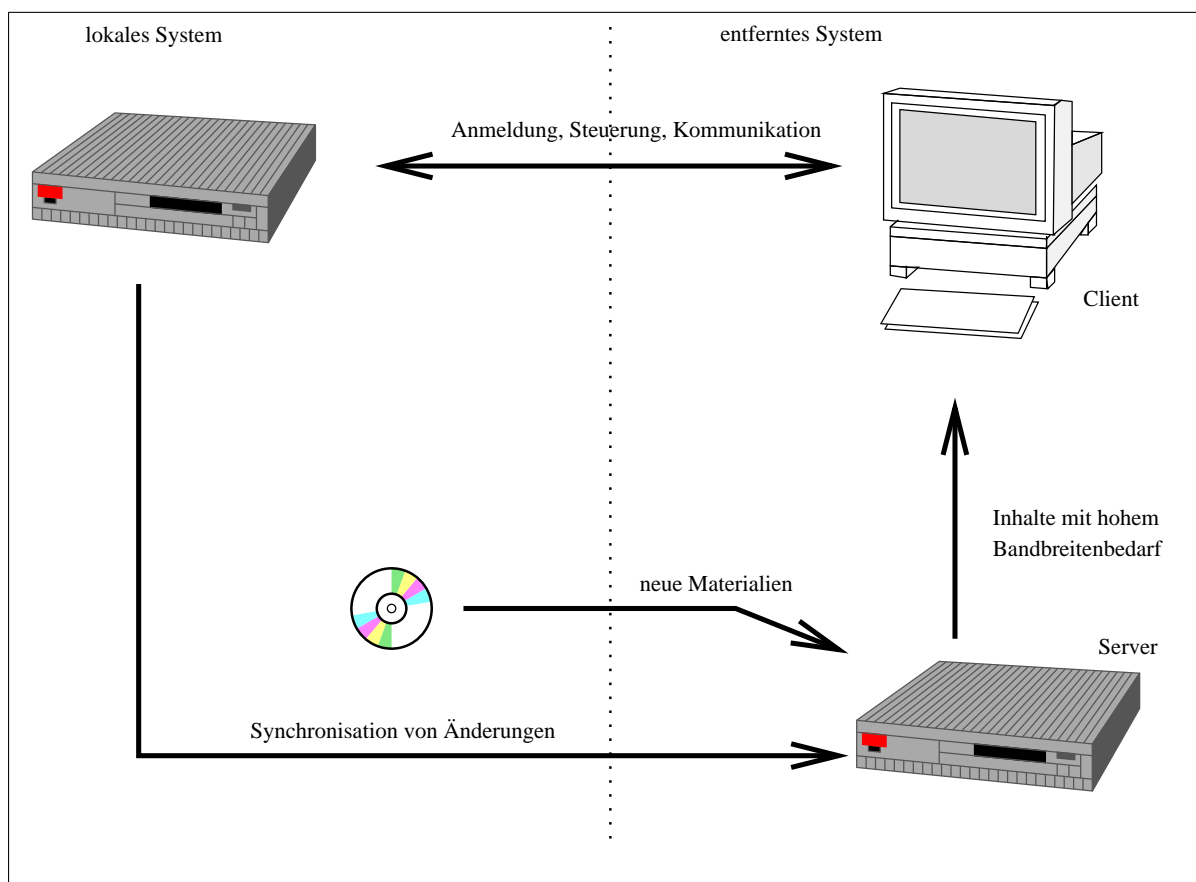


Abbildung 4.3: Datenaustausch mit zusätzlichem Server im entfernten Netz

Im Abschnitt 4.6.3.3 wird außerdem ein Verfahren beschrieben, welches es ermöglicht, Inhalte automatisiert priorisiert über ein Peer-to-peer Protokoll von lokalen Clients zu laden, um diese nicht über die langsamere Internetverbindung laden zu müssen.

4.6.1.3 Anbindung des Partners mit geringer Bandbreite

Steht als Internetanbindung der Partnerhochschule nur eine geringe Bandbreite zur Verfügung, sind folgende Szenarien denkbar:

- 1) Lokaler Betrieb der Lernplattform als Kommunikations- und Planungssystem, aber Ablage aller Inhalte auf den Clients oder einem Server an der Partnerhochschule (analog zu Abschnitt 4.6.1.2).
- 2) Zusätzliche Lernplattform auf einem Server der Partnerhochschule. Dabei kann eine Synchronisation über den Versand von Datenträgern stattfinden. Allerdings ist hier eine Nutzung der Plattform als Kommunikationsplattform zwischen beiden Standorten nur sehr eingeschränkt möglich, zumindest ein Austausch der Kommunikationsdaten über das Internet sollte realisiert werden.

4.6.1.4 Keine oder nur unzuverlässige Internetanbindung des Partners

Im Fall der fehlenden oder unzuverlässigen Internetanbindung des Partners kommt nur der Internet- unabhängige Betrieb in Frage. Dabei gibt es mehrere Möglichkeiten:

- 1) Export der Lernmaterialien vor Ort in ein Format, welches ohne LCMS verwendet werden kann, und Weitergabe dieser exportierten Daten. Dies hat den Nachteil, dass damit ein Großteil der interaktiven Funktionen, welche ein LCMS bereitstellt, verloren gehen.
- 2) Betrieb einer unabhängigen LCMS Plattform beim Partner. Die Lerninhalte werden in diesem Fall ebenfalls über den Export / Import über einen Datenträger übertragen. Dies ist nur möglich, wenn beim Partner ein entsprechender Server zur Verfügung steht, auf welchem die Plattform betrieben werden kann.
- 3) Weitergabe der gesamten, vorkonfigurierten Plattform auf einem Datenträger. Es ist, beispielsweise mit einer veränderten Knoppix-Installation möglich, die gesamte Lernplattform auf

einem USB Stick zu installieren und von diesem dann ein Linux-System zu booten, welches die Plattform unabhängig von der installierten Software des Clients startet und ohne Änderungen am Clientsystem auskommt. Alle Daten werden dabei auf dem USB Stick gespeichert. Hier lassen sich alle Funktionen der Plattform nutzen, ohne dass auf Seite des Partners ein zusätzlicher Aufwand entsteht (siehe Abschnitt 4.6.4).

4.6.2 Auswahl der Datenaustauschtechnologien

Im folgenden wird die Eignung einiger Technologien zum Datenaustausch zwischen verschiedenen Systemen für die Verwendung in LCMS Systemen beschrieben. Ebenfalls werden weitere, teilweise darauf aufbauende Techniken vorgestellt. Die verwendeten Basistechnologien wurden bereits im Kapitel 2.4 betrachtet.

4.6.2.1 RPC, CORBA

Die Technologien RPC und CORBA dienen der gemeinsamen Nutzung von Funktionen oder Objekten über das Netzwerk. Leider sind diese Techniken nicht plattform- und/oder implementierungsunabhängig, so dass es mit diesen Technologien nicht möglich ist, universelle Schnittstellen zwischen verschiedenen System zu erstellen. Da die von den LCMS verwendete Programmiersprache PHP diese Technologien nicht unterstützt, wäre eine Implementierung äußerst aufwendig, da ein Austausch nur über die Datenbank ohne Nutzung der API des LCMS erfolgen könnte.

4.6.2.2 XML-RPC, SOAP

XML-RPC und der Nachfolger SOAP eignen sich sehr gut, um Daten zwischen verschiedenen Systemen auszutauschen. Die Standards sind sehr weit verbreitet und werden von vielen Programmiersprachen direkt unterstützt, so dass ein systemübergreifender Datenaustausch damit sichergestellt werden kann. Zudem unterstützen einige LCMS bereits den externen Zugriff auf einige API-Funktionen über eine dieser Schnittstellen, was eine Anpassung und Erweiterung um Funktionen, welche zur Synchronisation zwischen mehreren LCMS genutzt werden

können, erleichtert. Aus diesen Gründen wurden für die Realisierung des Datenaustausches diese Techniken ausgewählt.

Darauf, und auf der bereits vorhandenen Schnittstelle in ILIAS aufbauend, wurden eine Suche und ein Synchronisationsmechanismus entworfen, welcher in Abschnitt 4.6.3.2 dargestellt wird.

4.6.2.3 Atom, RSS, ICE

Content Syndication Protokolle wie Atom, RSS und ICE dienen dem Austausch von Inhalten zwischen verschiedenen Systemen. Die Protokolle Atom und RSS sind dabei weit verbreitet, und werden beispielsweise eingesetzt, um Nachrichten von Webseiten, neue Foreneinträge und ähnliche Daten zwischen Servern oder zwischen Server und einem entsprechenden Client (RSS Reader) auszutauschen. Durch die Kombination mehrerer solcher Feeds ist es einfach, Informationen aus verschiedenen Quellen zusammenzufassen und übersichtlich darzustellen.

RSS und Atom bieten dabei aber protokollintern leider keine direkte Möglichkeit, die zu übertragenden Information auszuwählen. Diese werden immer vom Server nach festen Kriterien ausgewählt, so das beispielsweise immer die neuesten 10 Einträge in den Feed aufgenommen werden. Für den geplanten Einsatzzweck zur Synchronisation zwischen verschiedenen Systemen reicht dies aber nicht aus, da hier der Anfragende die Möglichkeit haben muss, Inhalte aufgrund der eigenen Konfiguration und den bereits vorhandenen Inhalten zu selektieren. Auch eine Authentifizierung ist nicht im Protokoll vorgesehen, sondern lässt sich nur auf höherer Ebene, beispielsweise HTTP, realisieren.

Das Protokoll ICE, welches in Version 2 auf SOAP basiert, bietet Möglichkeiten zur Authentifizierung, zum Abonnieren von Kanälen, und enthält Felder für Abrechnungszwecke. Auf Basis dieses Protokolls (allerdings noch Version 1) wurde in [Age06] ein Austausch von Lernmaterialien zwischen 2 ILIAS - Plattformen realisiert. Mit Hilfe des dort erstellten Scriptes ist es möglich, komplette Lerneinheiten, abhängig vom Datum der letzten Erstellung, zu übertragen. Allerdings wurden dabei nicht nur die Änderungen übertragen, sondern jeweils eine komplette Lerneinheit, die zuvor in ILIAS in das ILIAS-eigene Exportformat konvertiert werden musste.

4.6.2.4 Weitere Technologien

Für den Austausch von Lernmaterialien und die gemeinsame Bearbeitung gibt es noch weitere Möglichkeiten, teilweise unabhängig vom LCMS.

Versionsverwaltungssysteme Versionsverwaltungssysteme wie Subversion [PCSF04] oder das ältere CVS ([CVS], [Ced02]) eignen sich, neben ihrer Hauptfunktion, der Verwaltung verschiedener Softwareversionen und deren Änderungen, die man natürlich auch bei der LCMS Software selbst sehr gut einsetzen kann, um eigene Änderungen mit neuen Versionen abzugleichen, hervorragend zur Verwaltung von Text-, HTML- und XML basierten Lernmaterialien, insbesondere dann, wenn diese noch in Entstehung sind und von mehreren Personen bearbeitet werden. Da allerdings kein LCMS direkt eine Schnittstelle zu einem Versionsverwaltungssystem besitzt, ist es nötig, die entsprechenden Lernmaterialien entweder unabhängig vom LCMS zu erstellen oder aber geeignet zu Ex- und zu Importieren.

Dateisynchronisation Mit Hilfe von Programmen wie rsync ([Har04],[rsy]) ist es möglich, Änderungen an Dateien zwischen verschiedenen Systemen zu synchronisieren, auch über das Internet und zwischen verschiedenen Plattformen (wobei bei Windows-Plattformen der Installationsaufwand relativ hoch ist). Dabei werden nur die Änderungen an den Dateien übertragen, was bei rsync auch für binäre Dateien funktioniert. Damit lassen sich auf einfache Art und Weise beispielsweise vom LCMS angezeigte Dateidreier, welche als Dreierstruktur 1:1 im Dateisystem liegen und keinen Verweis in der Datenbank haben, auf den aktuellen Stand bringen. Dies ist insbesondere beim LCMS Moodle möglich. Existieren allerdings Verweise in der Datenbank, welche darüber nicht aktualisiert werden, werden neue Dateien vom LCMS nicht angezeigt. ILIAS speichert außerdem eine eigene Version der Datei in der Datenbank, welche bei externer Aktualisierung der Datei nicht aktualisiert wird, so dass dies hier nur bedingt geeignet ist.

Peer to Peer Datenaustausch Mit Hilfe von Peer to Peer Technologien, wie beispielsweise BitTorrent [bit] ist es möglich, umfangreiche Lernmaterialien, wie beispielsweise Videos, zur Entlastung des Servers statt vom LCMS Server direkt zwischen den Clients austauschen zu

lassen. LCMS-unabhängig ist es dabei möglich, im LCMS nur die entsprechende Steuerdatei (*.torrent) bereitzustellen sowie die entsprechende Infrastruktur (Tracker, Seeder), so dass die so bereitgestellten Dateien nur dann komplett vom eigenen Server geladen werden, wenn kein anderer Client diese zur Verfügung stellt.

4.6.3 Erweiterte Datenaustauschfunktionen

Die in Abschnitt 4.6.2 ausgewählte Technologie wurde um eigene Funktionen erweitert, um einen automatisierten Austausch von Daten für den Einsatz im Bildungsexport zu ermöglichen. Die einzelnen Erweiterungen werden im folgenden dargestellt.

4.6.3.1 Systemübergreifende Suchfunktionen

Auf Basis der API des LCMS ILIAS wurde eine XML-RPC Schnittstelle entwickelt, welche es ermöglicht, in einer Suchanfrage neben den Daten des eigenen LCMS auch den Datenbestand von weiteren LCMS zu durchsuchen. Dazu wird in eigenen LCMS eine Liste der anderen Systeme, welche auch durchsucht werden sollen, abgelegt. Diese Liste beinhaltet Links zu den jeweiligen XML-RPC Schnittstellen der anderen Systeme. Bei Durchführung einer Suche wird die eingegebene Suchanfrage parallel zur lokalen Suche auch über XML-RPC an die anderen Systeme weitergegeben. Gibt es dort Treffer, werden in der Antwort die Metadaten für die Inhalte des anderen Systems sowie ein Link zum Zugriff auf diese Daten zurückübermittelt, welche dann entsprechend in den Suchergebnissen des lokalen Systems mit angezeigt werden. Diese Suchfunktion wurde in der Diplomarbeit von Herrn Rakovskiy [Rak05] genau spezifiziert und als Beispiel implementiert.

4.6.3.2 Synchronisation von Änderungen über SOAP

Auf der Basis von SOAP und ausgehend von den Arbeiten von [Age06] und [Rak05] wurde eine Schnittstelle zum direkten Synchronisieren von Änderungen entworfen. Diese besteht aus einer Erweiterung der SOAP-API von ILIAS und einem Client zur Durchführung der Synchronisation (siehe Abbildung 4.4). Die Realisierung dieser Schnittstelle wird detailliert im Abschnitt

5.2 beschrieben. Dabei wurde auf Basis des Datenmodells von ILIAS und der ILIAS-API gearbeitet. Dies ermöglicht die Synchronisation zwischen mehreren ILIAS-Plattformen, erlaubt aufgrund der Ausnutzung der ILIAS-spezifischen Eigenschaften leider keine direkte Portierung auf andere Lernplattformen.

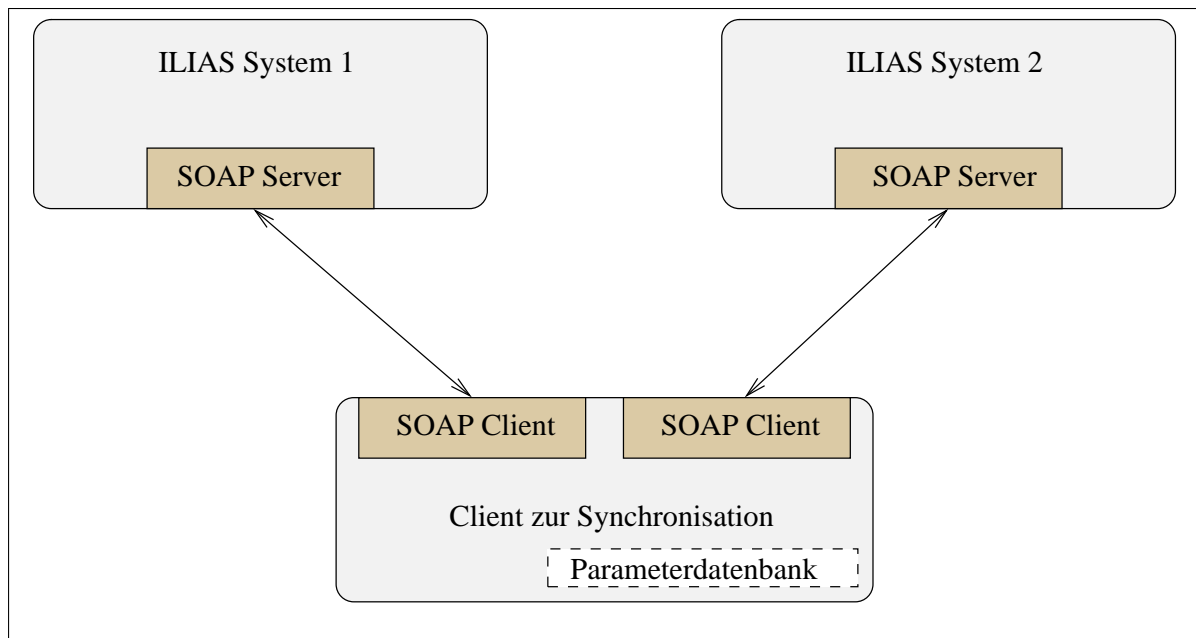


Abbildung 4.4: Aufbau der Synchronisation über SOAP

4.6.3.3 Übertragung von großen Dateien über Peer to Peer Netze

In [Tar08] wird eine Möglichkeit erläutert, die Übertragung von großen Dateien über Peer to Peer Netze weitergehend zu automatisieren und über einen Proxy-Server, welcher einen entsprechenden Peer to Peer Client enthält, dies ohne direkte Einbindung der Steuerdateien unabhängig vom bereits angelegten Lehrmaterial zu ermöglichen. Dabei wird von einem Server, dem Tracker eine Liste der Dateien vorgegeben, welche in den Peer to Peer Download aufgenommen werden wollen. Ein lokaler HTTP-Proxy, welcher gleichzeitig Mitglied im Peer to Peer Netz ist, vergleicht die Anfrage mit der Liste dieser Dateien und im positiven Fall wird die Datei vorrangig von anderen Clients geladen, bei welchen ebenfalls dieser Proxy eingesetzt wird. Nur wenn kein anderer Client verfügbar ist, wird die Datei vom Originalserver angefordert. Dieser Mechanismus ist transparent für Benutzer, die diesen Proxy nicht einsetzen, da in diesem Fall dann direkt auf den Originalserver zugegriffen wird.

4.6.4 Mobile Serversysteme

Um auch dort E-Learning zu ermöglichen, wo keine Internetverbindung oder nur eine Internetverbindung mit geringer Bandbreite verfügbar ist, allerdings ein lokales Netz existiert, wurden Verfahren erarbeitet, die es ermöglichen, einen E-Learning Server ohne Installation direkt von einem mobilen Datenträger (USB Stick, CD-ROM) zu starten. Dabei wurden mehrere Varianten betrachtet:

- Start der Lernumgebung innerhalb eines vorhandenen Windows-Systems
- Booten einer Lernumgebung direkt vom USB Stick

Bei beiden Varianten soll sichergestellt werden, dass das System des Hostrechners nicht geändert wird, also keine Dateien auf dem Hostsystem modifiziert werden, und alle anfallenden Daten auf den USB Stick geschrieben werden.

4.6.5 Weitere Anwendungen

Neben dem direkten Austausch von Lerninhalten zwischen Partnerhochschulen kann eine entsprechende Schnittstelle genutzt werden, um Lerninhalte an öffentliche Verzeichnisse zu übermitteln und ggfs. von diesen abzurufen. In einem solchen Verzeichnis, wie beispielsweise dem Bildungsportal Thüringen [BPT], werden Metadaten zu Lehrinhalten von verschiedenen Einrichtungen gesammelt und zur Recherche bereitgestellt. Da das Bildungsportal Thüringen Metadaten im LOM Format verarbeiten kann, sollte es mit wenigen Änderungen möglich sein, die Datenaustauschschnittstelle auch für automatische Veröffentlichungen der Lehrinhalte in einem Verzeichnis einzusetzen.

4.6.6 Abrechnung und Payment-Funktionen

ILIAS unterstützt bereits die Verwaltung von Kursen, welche nur gegen Bezahlung zugänglich sind. Allerdings erlauben die vorhandenen Funktionen nicht die Einbindung beliebiger, HTTP-basierter Zahlungssysteme. Daher wurden diese im Rahmen der Arbeit um eine zusätzliche Klasse erweitert, die als Basis für HTTP basierte Zahlungssysteme dient. Darauf aufbauend

wurde das Zahlungssystem „paybest“ als Beispiel mit Hilfe dieser Klasse eingebunden. Dies wird im Abschnitt 5.6 detailliert dargestellt.

Kapitel 5

Anwendungsbeispiele

In diesem Kapitel werden für einige im Kapitel 4 vorgestellte Varianten die konkreten technischen Realisierungen mit Darstellung der verwendeten Klassen und Auszügen aus den Quelltexten dargestellt.

5.1 Schnittstelle zum Austausch von Metadaten

Im Folgenden wird eine Schnittstelle beschrieben, welche den automatisierten Austausch von Lerneinheiten ermöglichen soll. Damit soll eine einfachere Wiederverwendbarkeit ermöglicht werden. Die Schnittstelle basiert auf der Diplomarbeit von Herrn Rakowsky [Rak05], allerdings wurde diese, im Gegensatz zur Arbeit von Herrn Rakowsky, auf Basis der bereits bestehenden ILIAS SOAP Infrastruktur erstellt.

Die ILIAS SOAP Schnittstelle enthält bereits Methoden zur Suche nach Objekten (*search-Objects*) sowie zum Auflisten aller Objekte (*getXMLTree*), welche sich auch auf einzelne Objekttypen, beispielsweise Lerneinheiten, beschränken lassen. Die Rückgabewerte dieser Funktionen sind XML-Daten nach der ILIAS-DTD, welche für jedes Objekt Identifikation, Titel, Beschreibung, Erstellungs- und Updatedatum sowie weitere Daten enthalten.

Die ILIAS SOAP API wurde um die Funktion *getMetaDataByRefId* ergänzt, welche es jetzt ermöglicht, zu einem Objekt mit bekannter ID die kompletten Metadaten im LOM-Format abzufragen (Quelltext 5.1).

Die zurückgelieferten Daten sind dabei an die Rechte des Benutzers gebunden, welcher sich über die SOAP-Schnittstelle angemeldet hat, so dass sich über die globalen Rollen sehr einfach steuern lässt, für welche Objekte die Daten abfragbar sind.

```

        function getMetaDataByRefId($sid, $a_ref_id)
        {
// Prüfung der Authentifizierung
            if(!$this->__checkSession($sid))
            {
                return $this->__raiseError($this->sauth->getMessage(),
                                            $this->sauth->getMessageCode());
            }

// Prüfung, ob übergebene Referenz gültig
            if(!is_numeric($a_ref_id))
            {
                return $this->__raiseError('No valid reference id given. Please choose '.
                                            an existing reference id of an ILIAS object',
                                            'Client');
            }

            include_once './include/inc.header.php';
            global $rbacsystem;

// Laden des Objektes mit der angegebenen Referenz
            if(!$tmp_obj = ilObjectFactory::getInstanceByRefId($a_ref_id,false))
            {
                return $this->__raiseError('Cannot create object instance!', 'Server');
            }

// Prüfung, ob der Benutzer Lesezugriff auf das Objekt hat
            if(!$rbacsystem->checkAccess('read', $tmp_obj->getRefId()))
            {
                return $this->__raiseError('No permission to read object !', 'Client');
            }

// Export der Metadaten des Objektes als XML
            include_once './classes/class.ilXmlWriter.php';
            $xml_writer =& new ilXmlWriter();
            $tmp_obj->exportXMLMetaData($xml_writer);
            return $xml_writer->xmlDumpMem();
        }

```

Quelltext 5.1: Abfrage der Metadaten über die SOAP-Schnittstelle in ILIAS

Einem Client (beispielsweise einem Verzeichnisdienst) steht es damit offen, zuerst die Liste der Objekte eines bestimmten Typs abzufragen, anhand der Erstellungs- / Aktualisierungsdaten

eine Auswahl zu treffen, welche Objekte eingetragen und aktualisiert werden sollen, und danach anhand der IDs die konkreten Metadaten für diese Objekte abzufragen.

Ebenso ist es möglich, dass ein Client eine direkte Suchanfrage über SOAP an das ILIAS-System weitergibt, für die Ergebnisliste die entsprechenden Metadaten abrufen und gemeinsam mit einem Link zum entsprechenden Objekt darstellt.

5.2 Synchronisation von Inhalten zwischen mehreren ILIAS-Systemen

Zur Synchronisation von Lerninhalten zwischen mehreren ILIAS-Systemen wurde eine Schnittstelle auf Basis des SOAP-Protokolls erstellt. Dafür wurde der ILIAS-Eigene SOAP-Server um einige Funktionen erweitert sowie ein SOAP-Client für den Abruf und Import der Daten auf dem Remotesystem erstellt.

5.2.1 Erweiterung des ILIAS SOAP Servers

Der in ILIAS bereits vorhandene SOAP-Server musste um einige Funktionen erweitert werden. Analog zum Abschnitt 5.1 können die vorhandenen Lerneinheiten zwar mit Hilfe der Funktion *getXMLTree* ermittelt werden, allerdings liefert diese Funktion nur die Einträge, welche sich im Repository vom ILIAS befinden. Dies schließt die Inhalte der Lerneinheiten nicht mit ein, weshalb eine Funktion *getContentObjectXML* implementiert wurde, welche auf den bereits vorhandenen XML Export in ILIAS zurückgreift und den Inhalt der Lerneinheit als XML zurückgibt. In der gegenwärtig realisierten Version werden dabei allerdings binäre Daten, welche als Medienobjekt eingebunden wurden, nicht mit zurückgeliefert.

Um Änderungen auch wieder zurückschreiben zu können, wurde eine Funktion *setContentObjectXML* implementiert. Diese Funktion erwartet als Parameter eine XML-Datei mit dem zu ändernden Objekt. Dies können in der aktuell implementierten Version *StructureObject* oder *PageObject* Objekte sein. Zusätzlich wurden 2 Attribute ergänzt, welche die Art der Änderung für das Objekt definieren. Die Identifikation des Objektes erfolgt dabei jeweils über das in XML-Struktur enthaltende Tag *Identifier*, welches bei allen Operationen mit angegeben

sein muss.

- *changetype*=“delete” löscht das Objekt
- *changetype*=“change” ersetzt das Objekt komplett durch die übergebene Version
- *changetype*=“append” *changeentry*=“ObjectId” Fügt das als XML-Struktur übergebene Objekt als Unterobjekt am Ende des mit der ObjectId übergebenen Objektes ein. Die ObjectId ist dabei nur bei *StructureObject* erforderlich und muss der Identifier eines Lernmoduls oder eines *StructureObject* sein. Rückgabewert der Funktion ist die neu erstellte Id des Objektes.
- *changetype*=“add-before” *changeentry*=“ObjectId” Fügt das als XML-Struktur übergebene Objekt vor dem angegebenen Objekt ein. Mit dieser Funktion können *StructuredObject* Objekte, aber auch *PageObject* oder *PageAlias* Objekte eingefügt werden.

5.2.2 Client zur Synchronisation von 2 ILIAS Systemen

Es wurde ein Client realisiert, welcher mit Hilfe der im vorhergehenden Abschnitt definierten Funktionen die Textinhalte zwischen 2 ILIAS-System synchronisieren kann. Dieser Client benutzt selbst eine Datenbank, um Änderungen verfolgen zu können, da in ILIAS Inhalte, im Gegensatz zu Repository-Objekten, keine Änderungsinformationen beinhalten. Der Ablauf des Einrichtens einer Synchronisation sieht daher wie folgt aus:

- Initialer, manueller Export der Lerneinheit auf System 1 und Import in System 2 (damit werden auch Medienobjekte übertragen)
- Abfragen der ObjectId nach dem Namen des Lernobjektes auf beiden Seiten mit Hilfe des Clients oder direkt im ILIAS-System
- Einrichten der Synchronisation am Client
- (regelmäßige) Synchronisation der Inhalte

Beim Einrichten der Synchronisation erfolgt eine Abfrage der Strukturen auf beiden Systemen durch den Client, die initiale Ablage der Inhalte für beide Clients in der Datenbank des Clients

sowie das Erstellen des Mappings der ObjectIds zwischen beiden Systemen.

Bei der folgenden, regelmäßig durchgeführten Synchronisation werden in mehreren Schritten die geänderten, gelöschten und hinzugefügten Seiten- und Strukturobjekte ermittelt und diese Änderungen auf beiden Seiten abgeglichen. Dabei wird wie folgt vorgegangen:

- Einlesen der kompletten aktuellen XML-Struktur von beiden Systemen
- Ermitteln neuer, geänderter und gelöschter Objekte anhand der gespeicherten Versionen in der Datenbank
- Abgleich der Objekte, welche auf beiden Seiten geändert wurden, gemäß einer vorgegebenen Strategie (Priorisierung einer Seite oder aber manueller Abgleich)
- Verarbeitung von Änderungen an *PageObject* Objekten, speichern der später für die Einbindung in die Struktur erforderlichen neuen ObjectIds in der Mapping-Tabelle
- Verarbeiten von Änderungen an *StructuredObject* Objekten

Dies wird in Abbildung 5.1 dargestellt.

Der Client verarbeitet dabei auch mehrere Synchronisationspaare und hält die entsprechenden Daten in der Datenbank, so dass es selbstverständlich auch möglich ist, mehrere Lernobjekte zu übertragen.

5.3 Applet für interaktive Folien

Die Realisierung zahlreicher interaktiver Folien wurde mit Hilfe eines Java-Applets (Abbildung 5.2 durchgeführt. Dabei bildet das einen Rahmen zur Realisierung verschiedener Abläufe, welche durch weitere Java-Klassen und Steuerdateien vorgegeben werden.

5.3.1 Konzept

Das realisierte Applet sollte im Wesentlichen folgende Eigenschaften besitzen:

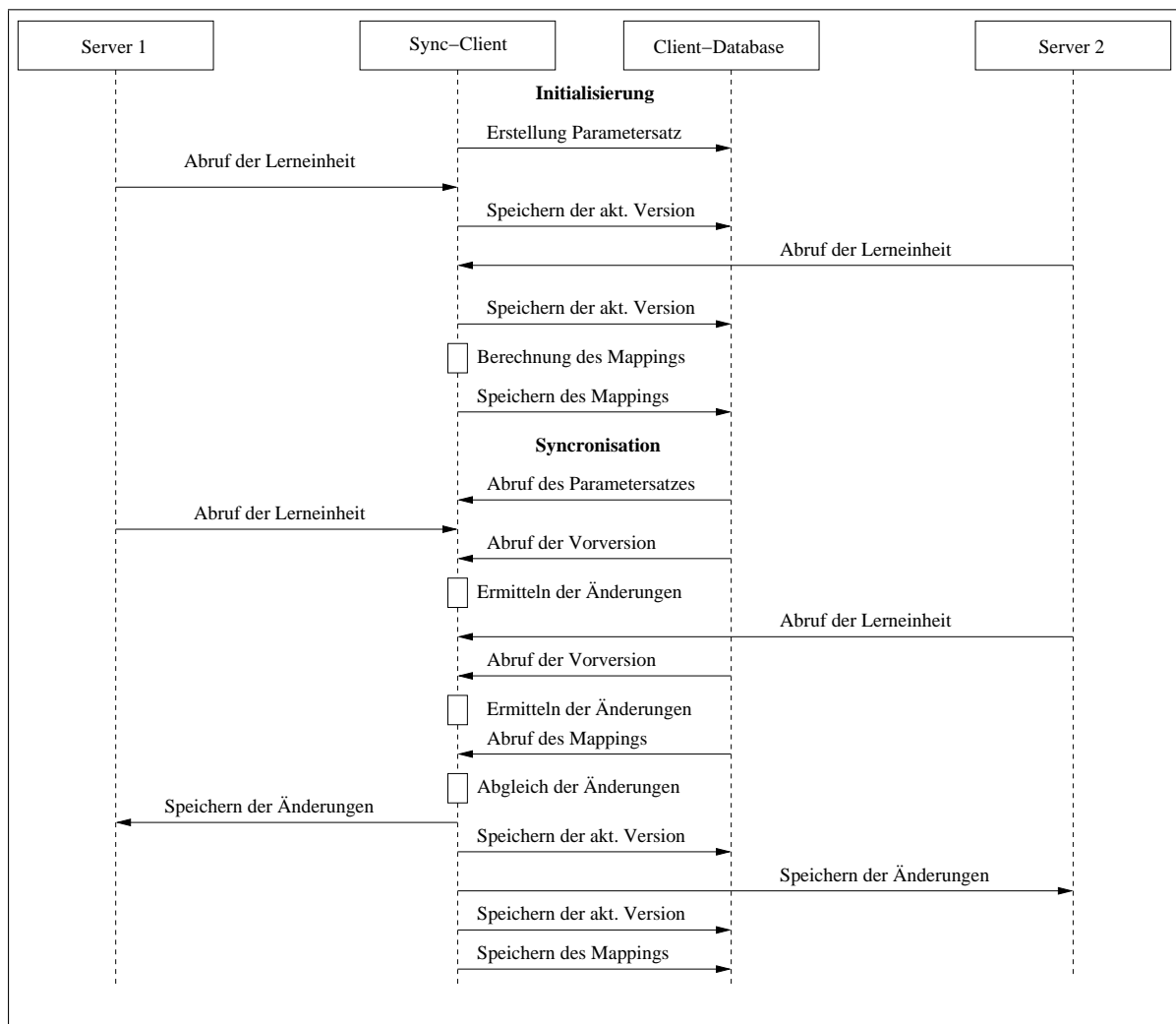


Abbildung 5.1: Ablauf der Datensynchronisation

- Möglichkeit der Umsetzung von verschiedensten Folien
- Einfache Erstellung neuer Folien
- Einfache Realisierung neuer interaktiver Elemente, z.B. Durchführung von Berechnungen
- Grundfunktionen wie das verschiedenfarbige Darstellen von Elementen, Erstellen von Basiselementen und einfachen, schrittweisen Abläufen, etc. ohne Programmierung möglich
- Externe Steuerung durch Parameterübergabe möglich

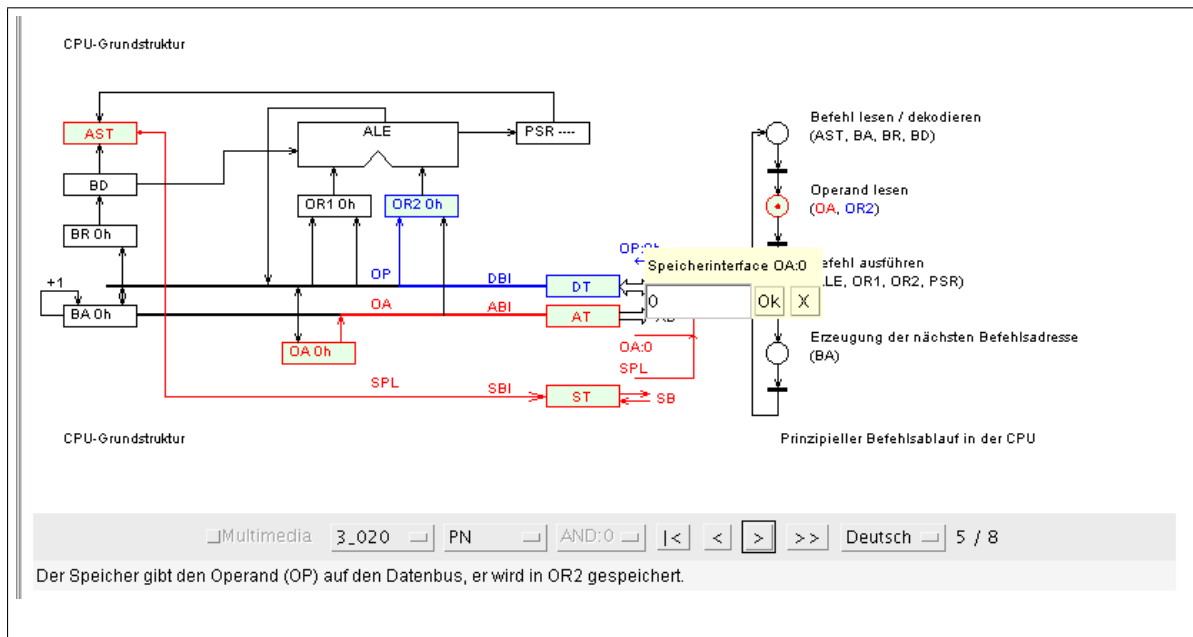


Abbildung 5.2: Applet für interaktive Folien

- Mehrsprachigkeit
- kontextabhängige Wiedergabe von Audio- und Videodateien

Um diese Eigenschaften umzusetzen, wurde eine abstrakte Klasse *Element* definiert, welche als Basisklasse für alle anzuzeigenden Elemente, wie Boxen, Pfeile, Texte, etc. dient. Neben der Definition der Schnittstelle aller abgeleiteten Klassen stellt diese Klasse auch alle Funktionen bereit, welche in den abgeleiteten Klassen zur Programmierung interaktiver Elemente benötigt werden, zum Beispiel der Zugriff auf andere Elemente, die Übersetzung von Texten, die Skalierung der einzelnen Elemente auf Bildschirmgröße, etc. Daneben stellt diese Klasse bereits Funktionen bereit, welche das farbliche Hervorheben von einzelnen Elementen ermöglichen.

Das in dieser Klasse definierte Interface, welches von allen abgeleiteten Klassen implementiert werden muß, beinhaltet neben Funktionen zur Darstellung der Elemente auch weitere Funktionen, welche ein einfaches Darstellen von verschiedenen Parametern ermöglichen (z.B. Registerinhalte), welche das Erstellen der Elemente durch das Bereitstellen von Parameterinformationen unterstützen, und weitere Funktionen.

Zahlreiche Klassen zur Darstellung von grundlegenden Elementen, wie zum Beispiel Boxen, Pfeilen, Bussystemen und Texten wurden ausgehend von dieser Klasse realisiert. Ausführliche Informationen zu diesen Klassen sind in Abschnitt 5.3.3 zu finden.

Die einzelnen Elemente werden vom Applet dynamisch geladen und verwaltet. Welche Klassen mit welchen Parametern instanziiert werden, wird dabei einer Beschreibungsdatei entnommen. Diese Datei, welche im Abschnitt 5.3.5 beschrieben ist, enthält darüber hinaus Informationen zum Ablauf während der Simulation, verfügbare Sprachen, Texte in verschiedenen Sprachen, Verweise auf abzuspielende Mediendateien und weitere Daten.

Neben diesen Elementen besteht das Applet aus folgenden Klassen:

- **SimApplet**
Zentrale Steuerung und Koordination der einzelnen Bestandteile
- **Model**
Verwaltung der Struktur, der einzelnen Instanzen der Elementklassen
- **SimulationEngine**
Verwaltung der Ablaufbeschreibung(en) und auszuführenden Aktionen
- **BundleSupport**
Verwaltung der Texte und Medienbeschreibungen für jede Sprache
- **MediaManager**
Steuerung der Wiedergabe von Multimediadateien

Eine Übersicht der Klassen ist in Abbildung 5.3 dargestellt. Die einzelnen Klassen werden im folgenden detailliert dargestellt.

5.3.2 Funktionsweise

5.3.2.1 Initialisierung / Struktur

Nach dem Laden des Applets und Auswerten der übergebenen Parameter wird die Beschreibungsdatei eingelesen. Die Datei ist in mehrere Abschnitte aufgeteilt. Im ersten Teil wird die

5.3.2.2 Simulation

Für jedes Modell lassen sich verschiedene Abläufe definieren. Ein solcher Ablauf besteht aus einem Script mit Anweisungen, welche für jeden Simulationsschritt folgende Befehle enthalten können:

- Hervorhebung von Elementen
- Auslösen von Aktionen, zum Beispiel Berechnungen (die Berechnungen selbst werden direkt als Java-Quelltext der jeweiligen Elementklasse definiert)
- Einfügen von Texten zur Erläuterung in die Grafik
- Festlegen des Erläuterungstextes in der Statusleiste und der abzuspielenden Mediendatei

5.3.3 Element* - Klassen

Die *Element** - Klassen, welche bereits im Abschnitt 5.3.1 erwähnt wurden, definieren die einzelnen darzustellenden Elemente. Dabei erben diese Klassen alle von der Klasse *Element*, welche einerseits eine Reihe abstrakte Funktionen definiert, welche von der jeweiligen Darstellungsklasse implementiert werden müssen, und andererseits Funktionen zum Zugriff auf andere Teile des Applets zur Verfügung stellt (Siehe Abbildung 5.4).

Folgende Methoden sind als Schnittstelle zum Applet in *Element* definiert und müssen von den einzelnen Klassen implementiert werden:

- **void init(SimApplet app, String Parameter)**

Diese Methode wird bei Erzeugung einer neuen Instanz mit den in der Beschreibungsdatei angegebenen Parametern aufgerufen. Die übergebenen Parameter sollten hier ausgewertet und gespeichert werden.

- **String getName()**

Diese Funktion muß einen für die Instanz eindeutigen Namen liefern. Dieser wird meist aus dem übergebenen Instanzparametern gebildet.

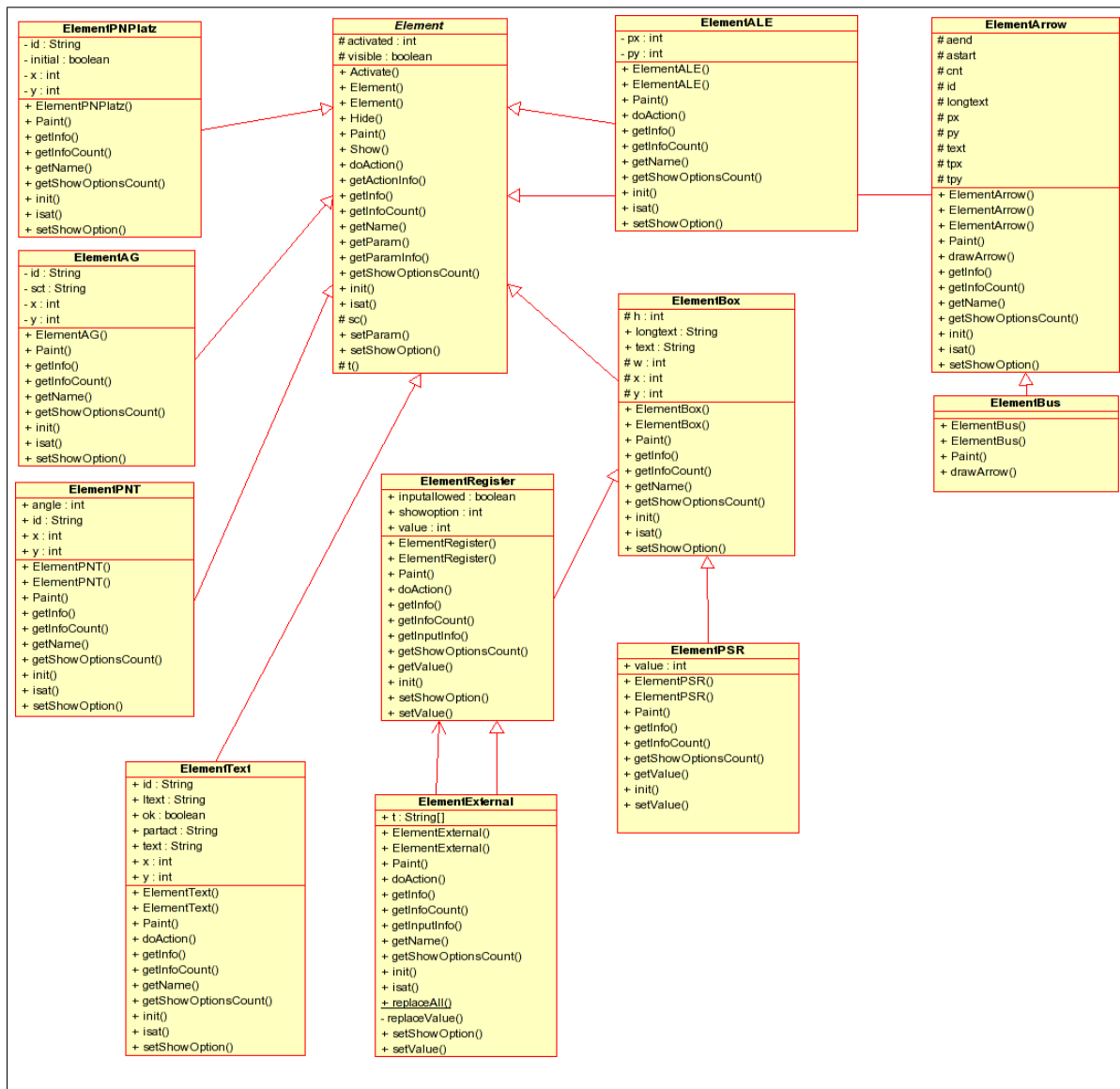


Abbildung 5.4: Applet für interaktive Folien: Klassendiagramm Elementklassen

- **void Activate(int Status)**

Diese Methode wird während der Simulation aufgerufen, wenn eine entsprechende Anweisung in der Beschreibungsdatei vorkommt. Ausgehend vom übergebenen Wert sollte hier die Farbe oder Art der Darstellung geändert werden.

- **void doAction(String Parameter)**

Diese Methode wird während der Simulation aufgerufen, wenn eine entsprechende Anweisung in der Beschreibungsdatei vorkommt. Ausgehend vom übergebenen Parameter

können hier Berechnungen durchgeführt und Werte anderer Elemente geändert werden. Durch die Implementierung dieser Funktion wird es möglich, Interaktion mit dem Benutzer zu integrieren. Der übergebene Parameter wird in der Beschreibungsdatei definiert und ist frei wählbar, so dass verschiedene Aktionen, ggfs. auch mit weiteren Parametern, definiert und während der Simulation ausgeführt werden können.

- **int getShowOptionCount(), void setShowOption(int option)**

Sollte die Klasse mehrere Darstellungsoptionen haben, so ist mit diesen Funktionen die Umschaltung der Darstellungsoptionen zu realisieren.

- **int getInfoCount(), String getInfo(int option)**

Diese Funktionen bilden eine Schnittstelle zur Abfrage erweiterter Informationen, welche nicht direkt dargestellt werden. Hier können neben einer längeren Beschreibung des Elementes oder der Instanz auch weitere berechnete Werte oder ähnliche Informationen dargestellt werden. Diese werden dann in der Statusleiste oder einem getrennten Fenster dargestellt.

- **boolean isat(int x, int y)**

Über diese Funktion wird ermittelt, ob sich das Element an der als Parameter übergebenen Bildschirmposition befindet.

- **void Paint(Graphics g)**

Diese Funktion wird immer dann aufgerufen, wenn das Element neu gezeichnet werden muß.

Die folgenden Funktionen bilden eine Schnittstelle zum in Abschnitt 5.4 beschriebenen Editor:

- **String getParamInfo()**

Diese Funktion liefert Informationen über die erwarteten Instanzparameter der Klasse. Dies ermöglicht es dem Editor, entsprechende Felder zur Bearbeitung der Daten bereitzustellen.

- **String getActionInfo()**

Diese Funktion liefert Informationen über die verfügbaren Aktionen der Klasse. Diese werden vom Editor bei Definition einer Aktion verwendet.

- **String getParam()**

Liefert die aktuellen Instanzparameter.

- **String setParam()**

Setzt die Instanzparameter auf einen neuen Wert.

Mit Erstellung des Editors wurde für die Instanzparameter ein festes Format eingeführt, um eine Auswertung und Bearbeitung bei verschiedenen Klassen zu vereinfachen.

Folgende Funktionen werden durch die Element-Basisklasse bereitgestellt und können von den abgeleiteten Klassen verwendet werden:

- **int sc(int size)**

Diese Funktion dient zur Skalierung von Bildschirmkoordinaten. Um immer eine bildfüllende Darstellung zu erreichen, werden die einzelnen Elemente mit Hilfe dieser Funktion vor dem Zeichnen skaliert.

- **String t(String text)**

Diese Funktion dient zum Nachschlagen des übergebenen Textes in der Sprachdatei der aktuell eingestellten Sprache und liefert die Übersetzung des Textes in die aktuelle Sprache, soweit vorhanden.

- **Element elemByName(String name)**

Liefert das Element mit dem angegebenen Namen. Dienst zum Zugriff auf andere Elemente, um diese zum Beispiel in eine Berechnung einzubeziehen.

5.3.4 Text/Audio Kopplung mit Java Media Framework

Alle Beschreibungstexte werden über die Klasse BundleSupport nach einer vergebenen ID angefragt und dort jeweils aus einer Datei mit der angewählten Sprache nachgeschlagen. Zur Unterstützung einer Text/Audio Kopplung wurde diese Klasse um weitere Elemente erweitert.

Im Applet selbst wurde das Textelement, welches sonst Hinweistexte, das aktuelle Element oder die Beschreibung des aktuellen Simulationsschritts anzeigt, gegen ein eigenes Panel ausgetauscht. Dieses erfüllt die gleiche Funktion der Anzeige der Daten, erlaubt aber weitere

Interaktionsmöglichkeiten. Während der akustischen Wiedergabe eines Textes wird der jeweils abgespielte Textabschnitt farbig hervorgehoben. Auch ist es möglich, durch einen Klick in einen Textabschnitt direkt zu diesem Textabschnitt zu springen, d.h. die zugehörige Audiodatei abzuspielen.

Die Zuordnung von Texten und Audiosequenzen erfolgt in einer Textdatei, welche ähnlich aufgebaut ist wie die Dateien für die Unterstützung der Mehrsprachigkeit. Diese enthalten jedoch die zugehörige Audiodatei sowie ggfs. Start-, Ende- und mehrere Indexpositionen, um das Springen im Text zu ermöglichen.

Das eigentliche Abspielen von Audiosequenzen erfolgt mit Hilfe der Klassen `MediaManager` und `MediaPlayer`. `MediaManager` verwaltet dabei die entsprechenden Start- und Endpositionen und ist für die Darstellung der Texte verantwortlich. `MediaPlayer` steuert direkt den Player des Windows Media Frameworks und wertet über `Listener` dessen Status aus. Dieser Status wird ausgewertet, und beispielsweise im Fall des Endes der aktuellen Wiedergabe beim Abspielen einer in der Animation verwendeten Audiosequenz verwendet, um im Ablauf der Animation den folgenden Schritt auszulösen.

Der `MediaPlayer` ist grundsätzlich auch in der Lage, neben Audio auch Videodateien abzuspielen. Aufgrund von Bandbreitenbeschränkungen und des Aufwandes der Produktion von Videos wurde dies aber nicht im erstellten Lehrmaterial realisiert.

Neben der direkt im Applet eingebunden Klasse `MediaManager` wurde eine weitere Klasse `MediaManagerNet` mit dem gleichen Interface realisiert, welche die Steuerbefehle zum Abspielen der Dateien und die entsprechenden Rückmeldungen zu/von einem Player-Applet, welches auf einem anderen Rechner läuft, über das Netzwerk sendet und von diesem empfängt.

5.3.5 Beschreibungs- / Steuerdateien

Das Applet verwendet zum Aufbau der Darstellung, zur Beschreibung der Animation, zur Definition der Texte in verschiedenen Sprachen und zur Definition der einzubindenden Mediantclips Steuerdateien auf Textbasis. Folgende Dateien werden verwendet:

- *models.txt*

Diese Datei enthält die verfügbaren Modelle, jeweils ein Name pro Zeile.

- *[modellname].txt*

Diese Datei enthält 2 Abschnitte. Im ersten Abschnitt wird der Aufbau des aktuellen Modells gesteuert. Dort ist angegeben, welche Elemente mit welchen Parametern für dieses Modell instanziiert werden sollen. Dabei werden mehrere Varianten der Animation in einer Datei unterstützt.

Im 2. Abschnitt ist das Script zur Darstellung der Animation enthalten. Neben der Hervorhebung von Elementen können hier über entsprechende Befehle dynamisch Texte ein- und ausgeblendet sowie Berechnungen in den einzelnen Element*-Klassen gestartet werden.

- *[Modellname].BundlesDescription.txt* Diese Datei enthält Informationen zu verfügbaren Sprachen, in welchen Dateien die Definitionen für die jeweilige Sprache gespeichert sind, sowie deren Codierung. Ebenso sind hier die Namen der Definitionsdateien für das Abspielen von Mediendateien enthalten.
- *[Modellname].[Sprache].ddl.txt* Diese Datei enthält für jedes definierte Label den entsprechenden anzuzeigenden Text in jeweils einer Sprache.
- *[Modellname].[Sprache].Media.ddl.txt* Diese Datei enthält für jedes definierte Label einen Verweis auf die passende Mediandatei sowie ggfs. Start-, End- und Indexpositionen.

5.4 Editor zum Erstellen der Applet - Beschreibungsdateien

Zum Erstellen der Steuerdateien für das Applet wurde eine Java-Applikation entworfen (Bild 5.5), mit welcher diese Dateien jeweils für ein Modell auf einfache Art und Weise bearbeitet werden können. Diese Anwendung erlaubt die Definition aller vom Applet verwendeten Steuerdaten. Lediglich die Implementierung neuer Elemente und Berechnungen müssen extern über neue Java-Klassen erfolgen.

Die Editoranwendung verwendet die gleichen Element*-Klassen wie das Applet. Diese wurden um eine Funktion ergänzt, welche die notwendigen Parameter inkl. einer Beschreibung der Parameter zurückliefert. Dies ermöglicht eine übersichtliche Eingabe direkt im Editor, ohne

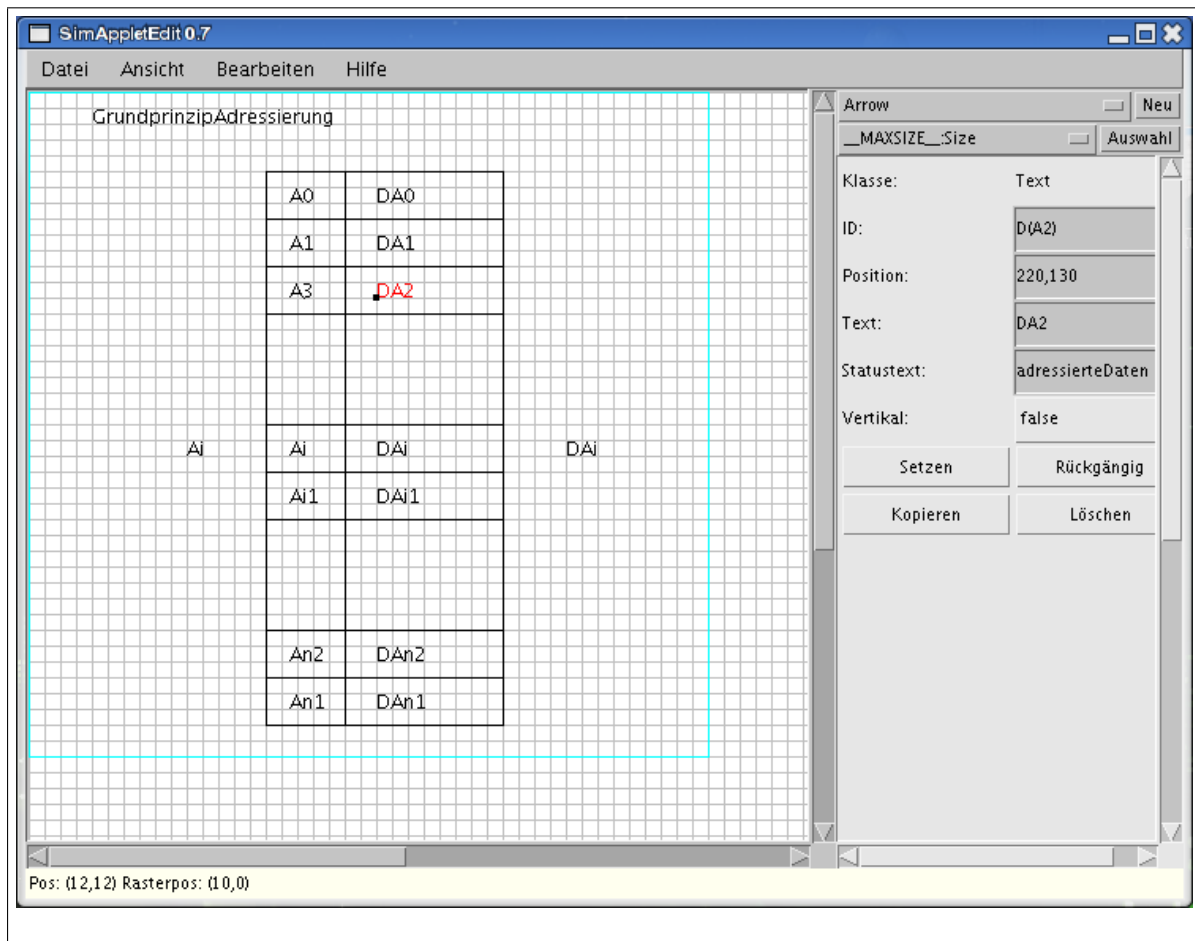


Abbildung 5.5: Editor für Applet-Beschreibungsdaten

die Syntax für jedes Element erst nachschlagen zu müssen. Koordinatenangaben und -listen können dabei direkt durch in Klick an die entsprechende Position im dargestellten Modell festgelegt werden, Hervorhebungen während der Simulation entsprechend durch Anklicken der einzelnen Elemente.

5.5 Applets zur Simulation spezialisierter Inhalte

Im Rahmen der Erstellung von Lehrmaterialien für das Fachgebiet Rechnerarchitekturen wurden weitere Applets entwickelt, um weitere Inhalte interaktiv zu präsentieren. Als weiteres Beispiel soll hier noch das Applet zur Darstellung und Simulation von Petri Netzen dargestellt werden. In diesem Applet wurde die Einbindung serverseitiger Funktionen für bestimmte

Funktionen realisiert, in diesem Fall für die Berechnung des Erreichbarkeitsgraphen (Abbildung 5.6).

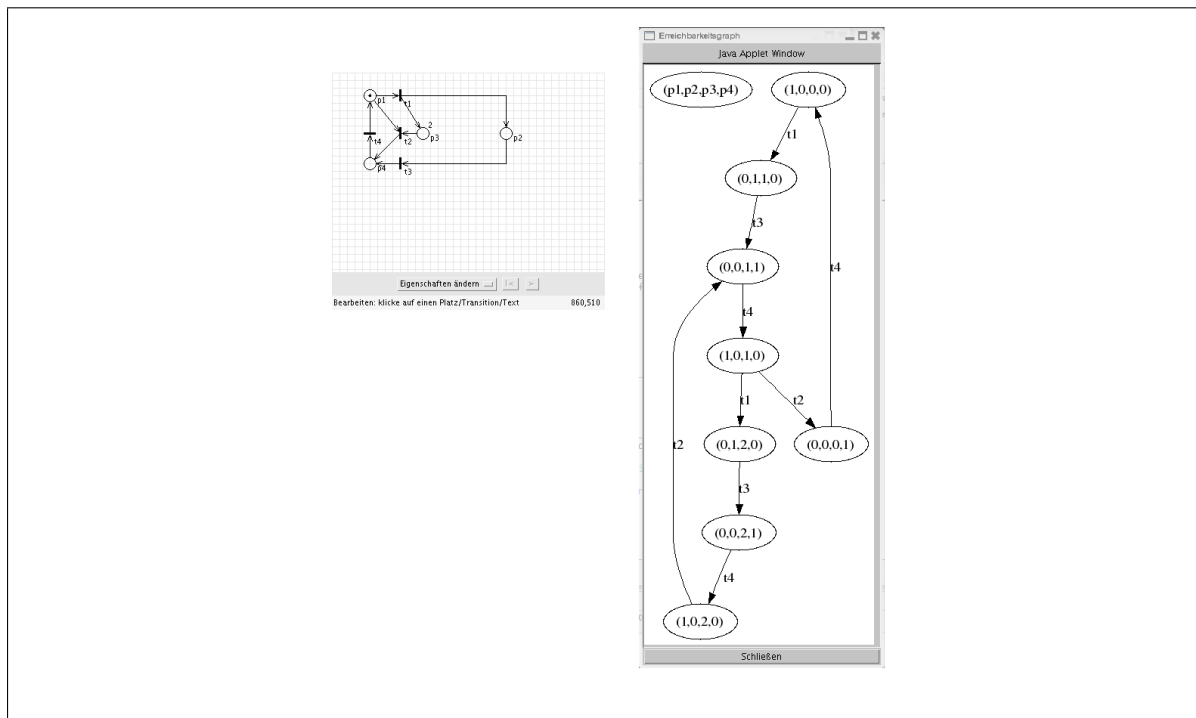


Abbildung 5.6: Applet Petri Netz Simulator mit Erreichbarkeitsgraph

Dabei wird auf dem Client im Applet eine Datenstruktur erzeugt, welche mittels eines HTTP POST Requests an den Server geschickt wird. Dieser verarbeitet die Daten und sendet als Antwort die Bilddaten für den Erreichbarkeitsgraphen, welche vom Applet daraufhin nur noch angezeigt werden. Dies ist in den Quelltexten 5.2 und 5.3 dargestellt. Quelltext 5.2 zeigt dabei die Übergabe der vorberechneten Daten an den Server. Das Ergebnis wird dabei als InputStream an die aufrufende Funktion zurückgegeben, welche im folgenden die Darstellung mit Hilfe der ImageIO Klassen von Java vornimmt.

Auf der Serverseite (Quelltext 5.3) werden die gesendeten Daten angenommen, entsprechend verarbeitet und, sofern die Berechnung erfolgreich war, das Ergebnis zurückgegeben.

```

try {
    // Aufbauen der Verbindung und setzen der Parameter
    URL url = new URL(app.getCodeBase(), "pndot.php");
    HttpURLConnection con = (HttpURLConnection) url.openConnection();
    con.setDoOutput(true);
    con.setDoInput(true);
    con.setRequestMethod("POST");
    con.setUseCaches(false);
    con.setRequestProperty("Content-Type",
                           "application/x-www-form-urlencoded");
    DataOutputStream out = new DataOutputStream(con.getOutputStream());

    // Vorbereiten und Ausgabe der Daten
    String data = "data=" + URLEncoder.encode(graph, "iso-8859-1") +
                  "&rnd=" + String.valueOf(Math.random());
    out.writeBytes(data);
    out.flush();
    out.close();

    // Rückgabe der Antwort als Stream, über welchen später die Grafik geladen wird
    return con.getInputStream();
}
catch (Exception ex) {
    System.out.println("Fehler beim Laden der Grafik");
    return null;
}

```

Quelltext 5.2: Auslagerung von Berechnungen auf den Server - Java-Applet (Client)

5.6 Paymentfunktionen in ILIAS

ILIAS enthält bereits die Möglichkeit, den Zugang zu Kursen nur gegen Zahlung einer Gebühr zu ermöglichen. Allerdings waren zum Zeitpunkt der Erstellung der Software nur die Zahlungsmethoden BMF (welches nicht allgemein nutzbar ist) und Rechnung implementiert. Aus diesem Grund wurde die Abrechnungsschnittstelle um URL-basierte Zahlungen erweitert, was am Beispiel von Paybest implementiert wurde.

Der grundlegende Ablauf einer Zahlung wurde bereits in Abschnitt 2.7, Abbildung 2.4 erläutert, in Abbildung 5.7 ist der Ablauf aus Nutzersicht dargestellt.

Der Button “Zur Kasse” ist dabei bereits in das Formular integriert, welches die Transak-


```

<?
// Enthält die übergebenen Daten
$data = $_POST["data"];
// Berechnung
$ergebnis = create_data_image($data);

if (!$ergebnis)
{
    Header("HTTP/1.x 500 Error");
} else {
    Header("Content-Type: image/png");
    echo $ergebnis;
}
?>

```

Quelltext 5.3: Auslagerung von Berechnungen auf den Server - serverseitige Berechnung

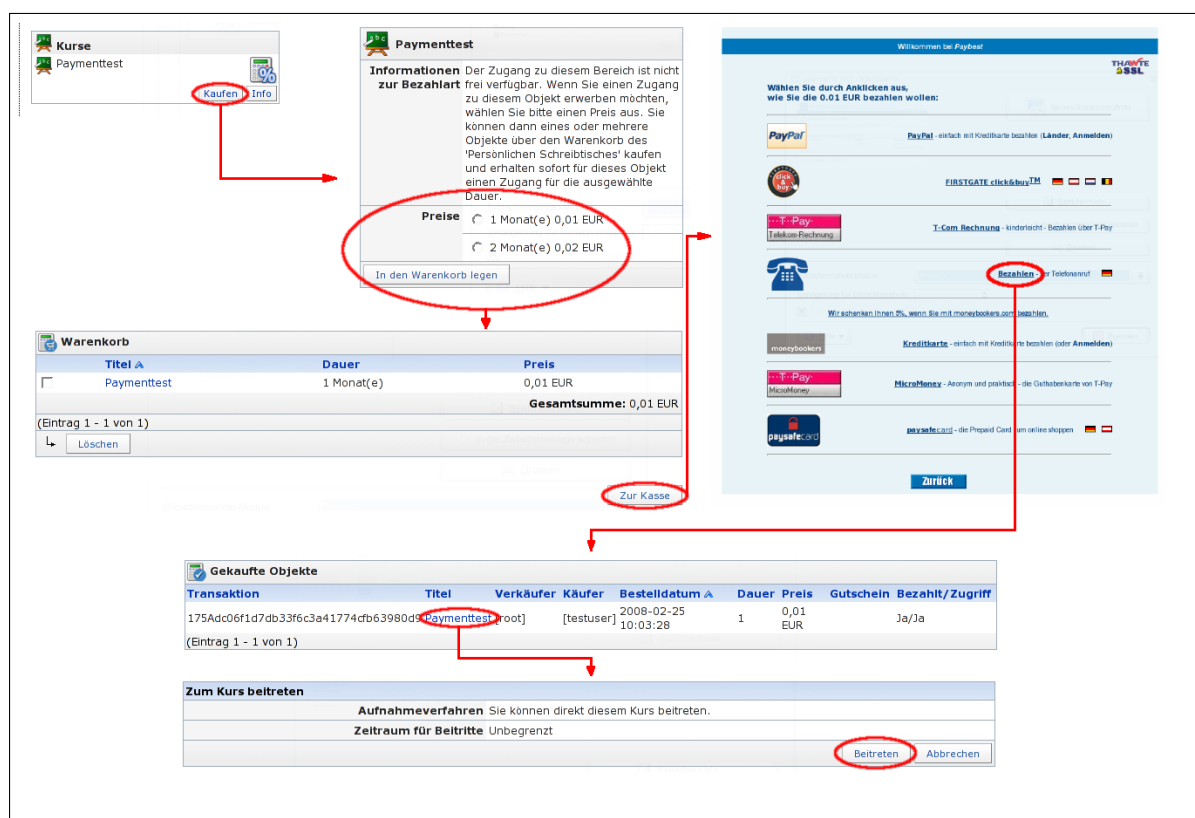


Abbildung 5.7: Ablauf des Bezahlvorgangs aus Benutzersicht

tionsdaten an den Zahlungsdienstleister übermittelt. Daher werden schon bei Erzeugen der Warenkorbansicht die entsprechenden Übergabeparameter und Signaturen erzeugt und direkt

an dieser Stelle eingebunden. Dazu wird in der Klasse *ilPaymentShoppingCartGUI* das entsprechende Formular aus einem Template geladen und mit den entsprechenden Parametern gefüllt (Quelltext 5.4). Die Daten werden dabei einer Instanz der Klasse *ilPurchasePaybest* entnommen, welche die generelle Verwaltung der Transaktionen und Berechnung der Parameter und Signaturen übernimmt. Alle feststehenden Daten, wie Server, Kundennummer, etc. des Zahlungsdienstleisters werden über das Objekt *paybestSettings* bzw. die daraus abgeleitete Datenstruktur *paybestConfig* abgefragt. Diese Werte werden vorab im Administrationsbereich von ILIAS erfasst (dazu wurde die Klasse *ilObjPaymentSettingsGUI* entsprechend erweitert).

```
$tpl =& new ilTemplate("./payment/templates/default/tpl.pay_shopping_cart_paybest.html",true,true);

// Setzen der Variablen im Template für die Übergabe der Daten
// Die nötigen Parameter werden dabei über die Methoden des paybest_obj berechnet
$tpl->setVariable("PAYBEST_HINT", $this->lng->txt('pay_hint_paybest'));
$tpl->setVariable("PAYBEST_INFO", $this->lng->txt('pay_info_paybest'));
$tpl->setVariable("GATEWAY_SCRIPT", $this->paybest_obj->paybestConfig["server"]);
$tpl->setVariable("CUSTOMER_ID", $this->paybest_obj->paybestConfig["customer_id"]);
$tpl->setVariable("SHOP_ID", $this->paybest_obj->paybestConfig["shop_id"]);
$tpl->setVariable("TARGET", "_self");

$tpl->setVariable("PRICE", $this->paybest_obj->getPrice());
$tpl->setVariable("PAYINGMODE", $this->paybest_obj->getPayingMode());
$this->paybest_obj->createSessionId();

$tpl->setVariable("SESSIONID", $this->paybest_obj->getSessionId());
$this->paybest_obj->setShopUrl($this->ctrl->getLinkTarget($this, 'finishPaybest'));
$tpl->setVariable("SHOPURL", $this->paybest_obj->getShopUrl());
$tpl->setVariable("PRODUCTS", $this->paybest_obj->getProducts());
$tpl->setVariable("HASH", $this->paybest_obj->getHash());

// Speichern der berechneten Werte in der Sitzung
$this->paybest_obj->saveToSession();
```

Quelltext 5.4: Übergabe der Zahlungsdaten im Formular

Zentrales Element ist die Klasse *ilPurchasePaybest*, welche die gesamte Paybest-spezifische Logik enthält und die Erzeugung und Prüfung der Signaturen (welche im Fall von Paybest mit MD5 gebildet werden) übernimmt. Als Beispiel wird hier die Prüfung der Rückgabewerte in Quelltext 5.5 dargestellt.

Diese Klasse wird von den angepassten ILIAS-Klassen zur Abwicklung der Zahlung verwendet, um die dienstspezifischen Berechnungen auszuführen. Im Fall einer Nutzung eines anderen Payment Providers muss daher nur diese Klasse und die Settings-Klasse zur Verwaltung der festen anbieterspezifischen Daten ausgetauscht werden.

```
function checkReturn()
{
    // Paybest return POST Parameter
    $r_sessionId = $_POST["sessionId"];
    $r_products = $_POST["products"];
    $r_units = $_POST["units"];
    $r_result = $_POST["result"];
    $r_hash = $_POST["hash"];
    $result = 0;

    // Berechnen der Prüfsumme und Vergleich mit dem übergebenen Wert
    $calc_hash = md5($r_sessionId . $r_products . $r_units . $this->paybestConfig["key2"] . $r_result);
    if ($calc_hash != $r_hash)
    {
        $result = PAYBEST_ERROR_HASH;
    }
    // Prüfung weiterer Parameter der Transaktion
    if ($r_sessionId != $_SESSION["paybest"]["sessionId"])
    {
        $result = PAYBEST_ERROR_SESSION;
    }
    if ($r_products != $_SESSION["paybest"]["products"])
    {
        $result = PAYBEST_ERROR_PRODUCTS;
    }
    if ($r_result != 1)
    {
        $result = PAYBEST_ERROR_ABORT;
    }

    $this->sessionId = $_SESSION["paybest"]["sessionId"];
    unset($_SESSION["paybest"]);
    return $result;
}
```

Quelltext 5.5: Prüfung der Rückgabewerte nach Abschluss der Zahlung

5.7 Playerscripting zur Synchronisation von Video- / Audiostreams und Präsentationsdaten

Unabhängig von Java-Applets wurden verschiedene Techniken realisiert, um Videos der Vorlesungen und Seminare und die dazugehörigen Text- und Bildmaterialien synchron wiederzugeben.

5.7.1 Aufzeichnung von Folienwechseln und Synchronisation mit einem Videostream

Die erste Möglichkeit erlaubt es, aufgezeichnete Vorlesungen oder Seminare gemeinsam mit Ihren in Powerpoint vorliegenden Folien auf schnelle Art und Weise in eine ILIAS - Lerneinheit umzuwandeln. Ausgangspunkt ist dabei ein während der Veranstaltung aufgezeichnetes Video und die Powerpoint-Präsentation.

Als erste Komponente wurde ein VBA-Modul realisiert, welches in Powerpoint eingebunden wird. Während der Aufzeichnung der Vorlesung auf Video und der Wiedergabe einer Powerpoint-Präsentation werden mit Hilfe dieses Makros die relativen Zeiten der Folienwechsel erfasst und in einer Datei gespeichert. Dabei muss nur die Präsentation über eine neue Schaltfläche "Präsentation mit Zeitnahme" gestartet werden, alles Weitere läuft im Hintergrund ab. Dieses Script ist auszugsweise in Quelltext 5.6 dargestellt. Als weiteres Makro enthält das Modul die Möglichkeit, die Folien in durchnummerierte Grafikdateien zu exportieren.

Eine zweite Komponente wandelt diese Datei dann in asx-Steuerdateien für den eingebundenen Medienplayer um, es wird für jede Folie eine asx-Datei mit den entsprechenden Parametern Dateiname, Startzeitpunkt und Dauer erzeugt. Die URL zum eigentlichen (WMV-) Video ist dabei Parameter anzugeben, ebenso, falls erforderlich, ein Offset, der einen Abgleich zwischen dem Starts der Präsentation und dem Start der Videoaufzeichnung ermöglicht.

Die dritte Komponente erzeugt dann aus den exportierten Bildsequenzen und den asx-Dateien ein ILIAS-Lernmodul. Nach dem Import in ILIAS kann dieses bei Bedarf noch mit den ILIAS-eigenen Bearbeitungsfunktionen ergänzt werden.

```

Private Function GetActiveSlide(ByVal Wnd As Object) As Slide
    Dim Sld As Slide
    Select Case TypeName(Wnd)
        Case "DocumentWindow"
            Set Sld = Wnd.Selection.SlideRange(1)
        Case "SlideShowWindow"
            Set Sld = Wnd.View.Slide
    End Select
    Set GetActiveSlide = Sld
End Function

Private Sub LogText(text As String)
    If Filename <> "" Then
        Set fs = CreateObject("Scripting.FileSystemObject")
        Set a = fs.OpenTextFile(Filename, 8, True)
        a.WriteLine (DateDiff("s", StartTime, Now) & " " & text)
        a.Close
    End If
End Sub

Private Sub appevent_SlideShowBegin(ByVal Wnd As SlideShowWindow)
    StartTime = Now
    LogText ("Start " & Date$ & " " & Time$)
    LogText ("Show Start")
End Sub

Private Sub appevent_SlideShowNextSlide(ByVal Wnd As SlideShowWindow)
    LogText (GetActiveSlide(Wnd).SlideNumber)
End Sub

```

Quelltext 5.6: Aufzeichnung der Folienwechselzeiten in Powerpoint

Die zweite und dritte Komponente wurden dabei aufgrund der einfachen Verarbeitung von Texten in Perl realisiert. Aufgrund der Größe der Videosequenzen sind diese dabei nicht mit im Lernmodul enthalten, sondern werden nur über die asx-Dateien referenziert.

Das erstellte Lernmodul enthält auf jeder Seite jeweils einen eingebetteten Player für den entsprechenden Teil des Videos sowie die zugehörige Grafikdatei der Folie. Die Navigation erfolgt über die Seitennavigation in ILIAS oder direkt über das Inhaltsverzeichnis.

Der Vorteil dieses Verfahrens ist die schnelle, größtenteils automatische Erstellung der Lernmaterialien. Nachteile sind, dass ein automatischer Sprung zur nächsten Folie am Ende des

aktuellen Videoabschnitts mit dieser Technik leider nicht möglich ist. Auch sind die Angaben zum Folienwechsel nur auf die Sekunde genau, da Powerpoint im VBA keine genaueren Zeitangaben liefert.

5.7.2 Synchronisation über Callbacks in der Videodatei

Eine weitere Möglichkeit ist das Einbetten von Steuerbefehlen direkt in die Videodatei. Dabei werden entweder direkt URLs einem anderen Fenster aufgerufen, oder aber es werden Steuerbefehle an einen JavaScript Handler weitergegeben, welcher dann für die passende Aktion zuständig ist [WMPb]. Dazu muss man die Videodatei mit dem Windows Media File Editor aus dem Windows Media Encoder Paket öffnen und an den entsprechenden Stellen manuell die auszuführenden Befehle einfügen. Es ist zwar auch möglich, dort direkt URLs zu hinterlegen, das ist aber in Hinsicht auf eine Weitergabe der Präsentation zu unflexibel. Hinterlegte Befehle werden an eine JavaScript Callback-Funktion übergeben, welche für das Auswerten der Parameter (z.B. die Foliennummer) und das Ändern der Inhalte zuständig ist. Über diese Funktion lässt sich eine entsprechende relative Verlinkung, passend zum eingesetzten LCMS, realisieren.

Bei dieser Lösung steuert der Player die Anzeige der weiteren Elemente, so das die Navigation über das Abspielen der Videodatei realisiert wird. Über das Festlegen von Sprungmarken in der Videodatei ist dabei auch das Anspringen einzelner Abschnitte möglich.

Der größte Nachteil des Verfahrens ist die zeitaufwendige Nachbearbeitung des Videos im Windows Media File Editor. Außerdem funktionieren die Callback-Handler leider nur im Internet Explorer zuverlässig, so das man sich damit an eine Plattform und einen Browser bindet.

Kapitel 6

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde eine Liste von Anforderungen erstellt, welche für den Einsatz von LCMS im erlösorientierten Bildungsexport benötigt werden. Die 4 wichtigsten Anforderungen waren hierbei der automatische Austausch von Lehrinhalten, die Einbindung von kontextabhängigen Multimediasequenzen, die Realisierung von Zahlungs- und Abrechnungsfunktionen sowie Untersuchungen zur Sicherheit von Inhalten.

Anhand der aufgestellten Anforderungen wurden verschiedene Open Source LCMS verglichen, um festzustellen, welches der Systeme sich am Besten für diesen Einsatzzweck eignet. Dabei wurde festgestellt, dass alle Anforderungen von keinem System erfüllt werden. Aufgrund der guten Erweiterbarkeit, der übersichtlichen API und der meisten erfüllten Anforderungen wurde entschieden, ILIAS Open Source als LCMS zu verwenden und im weiteren Verlauf der Arbeit mit diesem System zu arbeiten.

Es folgte die Erweiterung des LCMS um die in den Anforderungen festgestellte Funktionalität. Größter Komplex war hierbei die Realisierung des Datenaustausches zwischen mehreren Systemen, da hier zuerst ein passendes Protokoll gefunden werden musste, um dies zu realisieren und es gleichzeitig auch nötig war, auf die internen Strukturen des LCMS zuzugreifen. Es wurde das Simple Object Access Protokoll (SOAP) für den Zugriff auf die Objekte des LCMS verwendet, und ein Client entwickelt, welcher die Synchronisation zwischen mehreren Systemen anhand von definierten Vorgaben erlaubt.

Die weiteren Anforderungen waren einfacher zu realisieren. Für die Einbindung von Multimedia in interaktive Anwendungen und Lerneinheiten wurden mehrere Verfahren entwickelt: Einerseits, sofern die Inhalte als Java-Applet vorliegen, die Erweiterung der Applets um Multimediafähigkeiten mit Hilfe des Java Media Frameworks und einiger Steuerklassen, andererseits eine Methode, um als Audio- oder Videostream geplante Vorlesungen schon während

der Aufzeichnung mit den Präsentationsdaten zu synchronisieren und im folgenden eine automatische Erstellung von E-Learning Materialien aus den Präsentationsdaten und den Video- oder Audiodaten zu ermöglichen.

Zahlungs- und Abrechnungsfunktionen sind in ILIAS bereits vorhanden, allerdings unterstützte ILIAS außer der Zahlung auf Rechnung (Offline) und einer nicht öffentlich einsetzbaren BMF Abrechnung keinerlei andere Zahlungsdienstleister. Dies wurde durch eine Erweiterung der entsprechenden Schnittstelle geändert, so dass sich jetzt beliebige Zahlungsdienstleister, welche ein HTTP-basiertes Verfahren zum Austausch der Daten einsetzen, mit ILIAS Nutzen lassen. Als Beispielanwendung wurde dies für den Zahlungsdienstleister Paybest realisiert.

Die Sicherheit von Inhalten wurde bereits bei der Auswahl des LCMS als wichtiges Kriterium beachtet, so dass hier innerhalb des LCMS keine weiteren Anpassungen nötig waren. Für einzelne Inhalte wurden dagegen verschiedene Verfahren betrachtet, wie die Verschlüsselung von Teilen der Daten und Bindung an die Hardware des Clients, als auch die Auslagerung von Berechnungen und Komponenten auf einen Server, der diese Funktionen dann nur autorisierten Clients zur Verfügung stellt.

Als mögliche Erweiterungen wären als Erstes der Ausbau der Synchronisationsschnittstelle zu nennen, so dass sich, neben Lerninhalten auch andere Objekte über diese Schnittstelle synchronisieren lassen. Insbesondere eine Übertragung von eingebundenen Medien wäre wünschenswert.

Im Weiteren wäre eine Einbindung des Clients zur Synchronisation direkt in das LCMS denkbar, so dass Verwaltung der Synchronisationseinstellungen direkt aus dem LCMS heraus erfolgen kann. Auch eine direkte Anzeige von nicht-lokalen Repositories mit Möglichkeiten zum Import oder zur direkten Einrichtung einer Synchronisation wäre denkbar.

Literaturverzeichnis

- [4FO] 4FO AG: *Einfach bezahlen mit Paybest*. <http://www.paybest.de>, Abruf: 07.02.2008
- [ADL] ADL TECHNICAL TEAM: *Sharable Content Object Reference Model (SCORM) 2004 3rd Edition Documentation Suite*. <http://www.adlnet.gov/scorm>, Abruf: 07.02.2008
- [Ado] ADOBE SYSTEMS INCORPORATED: *Adobe Flash CS2 Professional*. <http://www.adobe.com/products/flash/>, Abruf: 07.02.2008
- [AFH06] ALTMANN, Werner ; FRITZ, Rene ; HINDERINK, Daniel: *TYPO3. Enterprise Content Management*. Open Source Press, 2006. – ISBN 393751418X
- [Age] AGENTUR FEIG & PARTNER: *Marktübersicht und Produktvergleich von Content Management Systemen*. <http://contentmenager.de>, Abruf: 07.02.2008
- [Age06] AGEeva, Alina: *Austausch von digitalen Lehrinhalten mittels Content Syndication Protokollen*, Technische Universität Ilmenau, Diplomarbeit, 2006
- [BCK⁺] BRODSKY, Jay (Hrsg.) ; CARRER, Marco (Hrsg.) ; KENNEDY, Dianne (Hrsg.) ; KOGER, Daniel (Hrsg.) ; MARTIN, Richard (Hrsg.) ; POPKIN, Laird (Hrsg.): *The Information and Content Exchange (ICE) Protocol, Working Draft Version 2.0*. <http://xml.coverpages.org/ICEv20-WorkingDraft.pdf>, Abruf: 07.02.2008
- [bds] *Bundesdatenschutzgesetz*. http://www.gesetze-im-internet.de/bdsg_1990/, Abruf: 07.02.2008
- [BEK⁺] BOX, Don ; EHNEBUSKE, David ; KAKIVAYA, Gopal ; LAYMAN, Andrew ; MENDELSON, Noah ; NIELSEN, Henrik F. ; THATTE, Satish ; WINER, Dave: *Simple Object Access Protocol (SOAP) 1.1*. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>, Abruf: 07.02.2008

- [BHMH03] BAUMGARTNER, Peter ; HÄFELE, Hartmut ; MAIER-HÄFELE, Kornelia: *E-Learning Praxishandbuch : Auswahl von Lernplattformen ; Marktübersicht - Funktionen - Fachbegriffe*. Studienverlag, Innsbruck [u.a.], 2003. – ISBN 3-7065-1771-X
- [bit] *BitTorrent protocol specification*. <http://www.bittorrent.org/protocol.html>, Abruf: 07.02.2008
- [Bol] BOLDT, Harry: *eLAIX - Editor für ILIAS 3 Lerneinheiten*. <http://www.boldt-media.de/elaix-news.html>, Abruf: 07.02.2008
- [BPT] *Bildungsportal Thüringen*. <http://bildungsportal-thueringen.de>, Abruf: 07.02.2008
- [CCMW] CHRISTENSEN, Erik (Hrsg.) ; CURBERA, Francisco (Hrsg.) ; MEREDITH, Greg (Hrsg.) ; WEERAWARANA, Sanjiva (Hrsg.): *Web Services Description Language (WSDL) 1.1*. <http://www.w3.org/TR/wsdl>, Abruf: 07.02.2008
- [Ced02] CEDERQVIST, Per: *Version Management With CVS*. Network Theory Ltd., 2002. – ISBN 0954161718
- [CF07] COLE, Jason ; FOSTER, Helen: *Using Moodle: Teaching with the Popular Open Source Course Management System (Using)*. O'Reilly Media, Inc., 2007. – ISBN 059652918X
- [CHRR] CLEMENT, Luc ; HATELY, Andrew ; RIEGEN, Claus von ; ROGERS, Tony: *UDDI Version 3.0.2*. http://uddi.org/pubs/uddi_v3.htm, Abruf: 07.02.2008
- [CVS] *CVS - Open Source Version Control*. <http://www.nongnu.org/cvs/>, Abruf: 07.02.2008
- [DA99] DIERKS, T. ; ALLEN, C.: *The TLS Protocol Version 1.0*. RFC 2246 (Proposed Standard). <http://www.ietf.org/rfc/rfc2246.txt>. Version: Januar 1999 (Request for Comments)
- [dat05] DATA-QUEST GMBH: *Produktbroschüre Stud.IP für Hochschulen*. Version: 2005. http://www.studip.de/download/mappe_studip-el.pdf, Abruf: 07.02.2008

- [dot] .LRN Home. <http://gotlrn.org>, Abruf: 07.02.2008
- [DR06] DIERKS, T. ; RESCORLA, E.: *The Transport Layer Security (TLS) Protocol Version 1.1*. RFC 4346 (Proposed Standard). <http://www.ietf.org/rfc/rfc4346.txt>. Version: April 2006 (Request for Comments)
- [FAPT07] FLEISCHER, A. ; ANDREEV, R. ; PAVLOV, Y. ; TERZIEVA, V.: An approach to personalized learning: a technique of estimation of learner's preferences. In: *Computer science meets automation; Vol. 2*, 2007
- [FF06] FLEISCHER, Alexander ; FENGLER, Wolfgang: Moderne Entwicklungen der Rechnerarchitektur. In: *Interner Workshop, Moskauer Energetisches Institut (TU)* (2006)
- [FGM⁺99] FIELDING, R. ; GETTYS, J. ; MOGUL, J. ; FRYSTYK, H. ; MASINTER, L. ; LEACH, P. ; BERNERS-LEE, T.: *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616 (Draft Standard). <http://www.ietf.org/rfc/rfc2616.txt>. Version: Juni 1999 (Request for Comments)
- [Fuc07] FUCHS, Daniel: *Web Content Management Systeme*. Vdm Verlag Dr. Müller, 2007. – ISBN 3836408392
- [FWH07] FINCKE, Sabine ; WUTTKE, Heinz-Dietrich ; HELSPER, Sandra: Dienstleistungen und Dienste des Bildungsportals Th[uringen]. In: *11. Workshop Multimedia in Bildung und Wirtschaft*, 2007
- [Ger07] GERTSCH, Fredi: *Das Moodle 1.8-Praxisbuch*. Addison-Wesley, München, 2007. – ISBN 3827325145
- [Har04] HARE, Chris: Keeping Data in Sync::rsync - Hare describes rsync, explains how it works, and discusses the interface between the client and the server. In: *Sysadmin : the journal for UNIX systems administrators* (2004), Nr. 2004 (06), S. 23–28. – ISSN 1061–2688
- [HJS01] HOFMANN, Johann ; JOBST, Fritz ; SCHABENBERGER, Roland: *Programmieren mit COM und CORBA*. Hanser Fachbuch, 2001. – ISBN 3446214798

- [ILI] ILIAS TEAM: *ILIAS Open Source Dokumentation*. <http://www.ilias.de>, Abruf: 07.02.2008

- [IMS] IMS GLOBAL LEARNING CONSORTIUM: *IMS Question & Test Interoperability Specification*. <http://www.imsproject.org/question/>, Abruf: 07.02.2008

- [JB] JACOB, Björn ; BARNEBECK, Robert: *Systematische Auswahl einer Open Source Lernplattform*. http://innoteach.bfh.ch/webdoc/Auswahl_Lernplattform.pdf, Abruf: 07.02.2008

- [Kla02] KLAUE, Christian: *Metadaten über Lernobjekte*, Technische Universität Ilmenau, Diplomarbeit, 2002

- [Koc06] KOCH, Stefan: *JavaScript - Einführung, Programmierung und Referenz - inkl. Ajax*. Dpunkt Verlag, 2006. – ISBN 3898643956

- [LFL06a] LAURENT, Simon S. ; FITZGERALD, Michael ; LAURENT, Simon S.: *XML kurz und gut*. O'Reilly, 2006. – ISBN 3897215160

- [LFL06b] LAURENT, Simon S. ; FITZGERALD, Michael ; LAURENT, Simon S.: *XML kurz und gut*. O'Reilly, 2006. – ISBN 3897215160

- [LOM02] *Draft Standard for Learning Object Metadata*. Version: 2002. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, Abruf: 07.02.2008

- [ML06] MINTERT, Stefan ; LEISEGANG, Christoph: *Ajax. Grundlagen, Frameworks und Praxislösungen*. Dpunkt Verlag, 2006. – ISBN 3898644049

- [moo] *Moodle - A Free, Open Source Course Management System for Online Learning*. <http://www.moodle.org>, Abruf: 07.02.2008

- [Nie06] NIEDERMEIER, Stephan: *Cocoon 2 und Tomcat. XML-Publishing mit dem Open-Source-Framework*. Galileo Press, 2006. – ISBN 3898426564

- [NS05] NOTTINGHAM, M. ; SAYRE, R.: *The Atom Syndication Format*. RFC 4287 (Proposed Standard). <http://www.ietf.org/rfc/rfc4287.txt>. Version: Dezember 2005 (Request for Comments)

- [Nüt02] NÜTZEL, Dr.-Ing. J.: Virtuelle Waren bezahlbar machen. In: JANTKE, K.P. (Hrsg.) ; WITTIG, W.S. (Hrsg.) ; HERRMANN, J. (Hrsg.): *Von e-Learning bis e-Payment*, Akademische V.-G., 2002
- [Nüt05] NÜTZEL, Dr.-Ing. J.: *Die informatorischen Aspekte virtueller Güter und Waren*. Universitätsverlag Ilmenau, 2005. – ISBN 3939473049. – Habilitationsschrift
- [Obja] OBJECT MANAGEMENT GROUP: *CORBA Component Model Specification*. <http://www.omg.org/docs/formal/06-04-01.pdf>, Abruf: 07.02.2008
- [Objb] OBJECTLAB: *openIPMP open-source rights management*. <http://objectlab.com/clients/openipmp/index.htm>, Abruf: 07.02.2008
- [Pai06] PAINTER, Mark: *Elektronische Medien in der Hochschullehre - Metadaten und Zugriffsstrukturen*. Shaker, 2006. – ISBN 3832251529
- [Paya] PAYPAL: *Online - Bezahlungssystem PayPal*. <http://www.paypal.de>, Abruf: 07.02.2008
- [Payb] PAYPAL: *PayPal Developer Central*. https://www.paypal.com/IntegrationCenter/ic_pdnHome.html, Abruf: 07.02.2008
- [PCSF04] PILATO, C. M. ; COLLINS-SUSSMAN, Ben ; FITZPATRICK, Brian W.: *Version Control with Subversion*. O'Reilly Media, Inc., 2004. – ISBN 0596004486
- [Rak05] RAKOVSKIY, Pavel: *Entwurf einer universellen Schnittstelle für den Austausch digitaler Lerneinheiten über das Internet*, Technische Universität Ilmenau, Diplomarbeit, 2005
- [Res00] RESCORLA, Eric: *SSL and TLS: Designing and Building Secure Systems*. Addison-Wesley Professional, 2000. – ISBN 0201615983
- [RSS] *RSS 2.0 Specification*. <http://blogs.law.harvard.edu/tech/rss>, Abruf: 07.02.2008
- [rsy] *rsync documentation*. <http://rsync.samba.org/>, Abruf: 07.02.2008

- [Sch] SCHULMEISTER, Prof. Dr. R.: *Selektions- und Entscheidungskriterien für die Auswahl von Lernplattformen und Autorenwerkzeugen*. <http://www.izhd.uni-hamburg.de/pdfs/Plattformen.pdf>, Abruf: 07.02.2008
- [Sch06] SCHILDT, Herbert: *Java: The Complete Reference, Seventh Edition*. McGraw-Hill Osborne Media, 2006. – ISBN 0072263857
- [Sri95] SRINIVASAN, R.: *RPC: Remote Procedure Call Protocol Specification Version 2*. RFC 1831 (Proposed Standard). <http://www.ietf.org/rfc/rfc1831.txt>. Version: August 1995 (Request for Comments)
- [STK02] SNELL, James ; TIDWELL, Doug ; KULCHENKO, Pavel: *Webservice-Programmierung mit SOAP*. O'Reilly, 2002. – ISBN 3897211599
- [Sun] SUN MICROSYSTEMS, INC.: *Java Technology*. <http://java.sun.com/>, Abruf: 07.02.2008
- [Sun88] SUN MICROSYSTEMS: *RPC: Remote Procedure Call Protocol specification: Version 2*. RFC 1057 (Informational). <http://www.ietf.org/rfc/rfc1057.txt>. Version: Juni 1988 (Request for Comments)
- [Tar08] TARASOV, Mikhail: *Verteilung von Lehrmaterialien über P2P-Strukturen*, Technische Universität Ilmenau, Diplomarbeit, 2008
- [The] THE APACHE SOFTWARE FOUNDATION: *Lucene*. <http://lucene.apache.org>, Abruf: 07.02.2008
- [Tse06] TSELISHCHEV, Alexey: *Digitale Sicherheit und Rechteverwaltung beim Einsatz von E-Learning*, Technische Universität Ilmenau, Diplomarbeit, 2006
- [TYP] TYPO3 ASSOCIATION: *TYPO3 Content Management System*. <http://typo3.com>, Abruf: 07.02.2008
- [WCL⁺05] WEERAWARANA, Sanjiva ; CURBERA, Francisco ; LEYMAN, Frank ; STOREY, Tony ; FERGUSON, Donald F.: *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More*. Prentice Hall PTR, 2005. – ISBN 0131488740

- [Wes07] WESCHKALNIES, Nick: *Adobe Flash CS3*. Galileo Press, 2007. – ISBN 3836210649
- [WFF03] WUTTKE, Heinz-Dietrich ; FLEISCHER, Alexander ; FENGLER, Wolfgang: Class concept for Living Pictures. In: *World Multiconference on Systemics, Cybernetics and Informatics*, 2003
- [WMPa] *Microsoft Producer*. <http://www.microsoft.com/windows/windowsmedia/technologies/producer.msp>, Abruf: 07.02.2008
- [WMPb] *Windows Media Player 10 SDK*. <http://msdn2.microsoft.com/en-us/windowsmedia/bb190327.aspx>, Abruf: 07.02.2008
- [WOH⁺] WEBBER, Neil (Hrsg.) ; O’CONNELL, Conleth (Hrsg.) ; HUNT, Bruce (Hrsg.) ; LEVINE, Rick (Hrsg.) ; POPKIN, Laird (Hrsg.) ; LAROSE, Gord (Hrsg.): *The Information and Content Exchange (ICE) Protocol*. <http://www.w3.org/TR/NOTE-ice>
- [WSK⁺02] WUTTKE, Heinz-Dietrich ; SCHMIDT, Karsten ; KRATZ, Michael ; KÖHLER, Thomas ; UNKROTH, Andreas ; MARTENS, Jürgen: Metadaten für das Bildungsportal Thüringen. In: *Campus 2002 : die virtuelle Hochschule in der Konsolidierungsphase*, Waxmann, Münster u.a., 2002
- [Wut06] WUTTKE, Heinz-Dietrich: An applet-based approach to reuseable learning objects. In: *IASTED International Conference on Computers and Advanced Technology in Education*, ACTA Press, 2006